

# mvord: An R Package for Fitting Multivariate Ordinal Regression Models

Rainer Hirk

WU Wirtschaftsuniversität  
Wien

Kurt Hornik

WU Wirtschaftsuniversität  
Wien

Laura Vana

WU Wirtschaftsuniversität  
Wien

---

## Abstract

The R package **mvord** implements composite likelihood estimation in the class of multivariate ordinal regression models with probit and a logit link. A flexible modeling framework for multiple ordinal measurements on the same subject is set up, which takes into consideration the dependence among the multiple observations by employing different error structures. Heterogeneity in the error structure across the subjects can be accounted for by the package, which allows for covariate dependent error structures. In addition, regression coefficients and threshold parameters are varying across the multiple response dimensions in the default implementation. However, constraints can be defined by the user if a reduction of the parameter space is desired.

*Keywords:* composite likelihood, cross-sectional data, longitudinal data, multivariate ordinal regression model, R.

---

## 1. Introduction

The analysis of ordinal data is an important task in various areas of research. One of the most common settings is the modeling of preferences or opinions (on a scale from, say, *poor* to *very good* or *strongly disagree* to *strongly agree*). The scenarios involved range from psychology (e.g., aptitude and personality testing), marketing (e.g., consumer preferences research) and economics and finance (e.g., credit risk assessment for sovereigns or firms) to information retrieval (where documents are ranked by the user according to their relevance) and medical sciences (e.g., modelling of pain severity or cancer stages).

Most of these applications deal with correlated ordinal data, as typically multiple ordinal measurements or outcomes are available for a collection of subjects or objects (e.g., interviewees answering different questions, different raters assigning credit ratings to a firm, pain levels being recorded for patients repeatedly over a period of time, etc.). In such a multivariate setting, models which can deal with the correlation in the ordinal outcomes are desired. One possibility is to employ a *multivariate ordinal regression model* where the marginal distribution of the subject errors is assumed to be multivariate. Other options are the inclusion of random-effects in the ordinal regression model and conditional models (Fahrmeir and Tutz 2001).

Several ordinal regression models can be employed for the analysis of ordinal data, with cumulative link models being the most popular ones (see Tutz 2012; Christensen 2015a). Other

approaches include continuation-ratio or adjacent-category models (see [Agresti 2002, 2010](#)). Different packages to analyze and model ordinal data are available in R ([R Core Team 2017](#)): For univariate ordinal regression models with fixed effects the function `polr()` of the **MASS** package ([Venables and Ripley 2002](#)), the function `clm()` of the **ordinal** package ([Christensen 2015b](#)) which supports scale effects as well as nominal effects and the function `vglm()` of the **VGAM** package ([Yee 2010](#)). Another package which accounts for heteroskedasticity is package **oglmx** ([Carroll 2016](#)). The package **ordinalNet** ([Wurm, Rathouz, and Hanlon 2017](#)) offers tools for model selection by using an elastic net penalty. Moreover, ordinal logistic models can be fitted by the functions `lms()` and `orm()` in package **rms** ([Harrell Jr 2017](#)) and the `MCMCprobit()` function in package **MCMCpack** ([Martin, Quinn, and Park 2011](#)) which uses Markov Chain Monte Carlo methods to fit ordinal probit regression models.

An overview on ordinal regression models in other statistical software packages like Stata ([StataCorp. 2017](#)), SAS ([SAS Institute Inc. 2017](#)) or SPSS ([SPSS Inc. 2017](#)) is provided by [Liu \(2009\)](#). These software packages include the Stata procedure `OLOGIT`, the SAS procedure `PROC LOGISTIC` and the SPSS procedure `PLUM` which perform ordinal logistic regression models. The software procedure `PLUM` additionally includes other link functions like probit complementary log-log, cauchit and negative log-log. Ordinal models for multinomial data are available in the SAS package `PROC GENMOD`, while another implementation of ordinal logistic regression is available in JMP. In Python ([Python Software Foundation 2017](#)), package **mord** ([Pedregosa-Izquierdo 2015](#)) implements ordinal regression methods.

While there are sufficient software tools in R which deal with the univariate case, the ready-to-use packages for dealing with the multivariate case fall behind, mainly due to computational problems or lack of flexibility in the model specification. However, there are some R packages which support correlated ordinal data. One-dimensional normally distributed random effects in ordinal regression can be handled by the `clmm()` function of package **ordinal** ([Christensen 2015b](#)); multiple possibly correlated random effects are implemented in package **mixor** ([Hedeker, Archer, Nordgren, and Gibbons 2015](#)). Note that this package uses multi-dimensional quadrature methods and estimation becomes infeasible for increasing dimension of the random effects. Bayesian multilevel models for ordinal data are implemented in the package **brms** ([Bürkner 2017](#)). Multivariate ordered probit models, where the subject errors are assumed to follow a multivariate normal distribution with a general correlation matrix, can be estimated with package **PLordprob** ([Kenne Pagui, Canale, Genz, and Azzalini 2014](#)), which uses maximum composite likelihood methods estimation. This package works well for standard applications but lacks flexibility. For example, the number of levels of the ordinal responses needs to be equal across all dimensions, threshold and regression coefficients are the same for all multiple measurements and it does not account for missing observations in the outcome variable. None of these packages support at the time of writing covariate dependent error structures. A package which allows for different error structures in non-linear mixed effects models is package **nlme** ([Pinheiro, Bates, DebRoy, Sarkar, and R Core Team 2017](#)), even though models dealing with ordinal data are not supported.

The motivation of this package lies in a credit risk application, where multiple credit ratings are assigned by various credit rating agencies (CRAs) to firms over several years. CRAs have an important role in financial markets, as they deliver subjective assessments or opinions of an entity's creditworthiness, which are then used by the other players on the market, such as investors and regulators, in their decision making process. Entities are assigned to rating classes by CRAs on an ordinal scale by using both quantitative and qualitative criteria.

Ordinal credit ratings can be seen as a coarser version of an underlying continuous latent process, which measures creditworthiness. In the literature, this latent variable motivation has been used in various credit rating models (e.g., [Blume, Lim, and Mackinlay 1998](#); [Afonso, Gomes, and Rother 2009](#); [Alp 2013](#); [Reusens and Croux 2017](#)).

This setting is an example of an application where correlated ordinal data arises naturally. On the one hand, multiple ratings for one firm at the same point in time can be assumed to be correlated and on the other hand, given the longitudinal dimension of the data, for each rater, there is serial dependence in the ratings assigned over several periods. Moreover, aside from the need of a model class that can handle correlated ordinal data, additional flexibility is desired due to the following characteristics of the problem at hand: Firstly, there is heterogeneity in the rating methodology. Raters use different labeling as well as a different number of rating classes. Secondly, the credit risk measure employed in assessing creditworthiness can differ among raters (e.g., probability of default versus recovery in case of default), which leads to heterogeneity in the covariates, as raters might use different variables in their rating process and assign different importance to the variables employed. Thirdly, the data has missing values and is unbalanced, as firms can leave the data set before the end of the observation period due to various reasons such as default but also because of mergers and acquisitions, privatizations, etc., or ratings can be withdrawn. Moreover, there are missings in the multiple ratings, as not all firms are rated by all CRAs at each time point.

The paper and package **mvord** ([Hirk, Hornik, and Vana 2017b](#)) for R aim at providing a flexible framework for analyzing correlated ordinal data by means of the class of multivariate ordinal regression models. We offer the following features which (to the best of our knowledge) enhance the currently available software for multivariate ordinal regression models in R:

- i Different error structures are available (such as a general, an autoregressive order one, or an equicorrelation structure for the subject error);
- ii we also account for heterogeneity in the error structure among the subjects by allowing the use of subject-specific covariates in the specification of the error structure;
- iii we implement a multivariate logit link for the class of multivariate ordinal regression models;
- iv we allow for outcome specific threshold parameters;
- v we allow for outcome specific regression parameters;
- vi the user can impose further restrictions on the threshold and regression parameters in order to achieve a more parsimonious model (e.g., using one set of thresholds for all outcome dimensions);
- vii we offer the possibility to switch between different parameterizations, which are needed in ordinal models to ensure identifiability.

This paper is organized as follows: Section 2 provides an overview of the model class and the estimation procedure, including model specification and identifiability issues. Section 3 presents the main functions of the package. A description of the illustrative data sets used for exemplifying the functionalities of the package is given in Section 4. A couple of worked examples are given in Section 5. Section 6 concludes.

## 2. Model class and estimation

Multivariate ordinal regression models are an appropriate modeling choice when a vector of correlated ordinal response variables, together with covariates, is observed for each unit or subject in the sample. The response vector can be composed of different variables, i.e., multiple measurements on the same subject (e.g., different credit ratings assigned to a firm by different CRAs, different survey questions answered by an interviewee, etc.) or repeated measurements on the same variable at different time points.

### 2.1. Model specification

Let  $Y_{ij}$  denote the ordinal observation and  $\mathbf{x}_{ij}$  be a  $p$  dimensional vector of covariates for subject  $i$  and outcome  $j$ , where  $i = 1, \dots, n$  and  $j \in J_i$ , for  $J_i$  a subset of all available outcomes  $J$  in the data set. Moreover, we denote by  $q = |J|$  and  $q_i = |J_i|$  the number of elements in the set  $J$  and  $J_i$ , respectively. Following the cumulative link modeling approach (McCullagh 1980), the ordinal response  $Y_{ij}$  is assumed to be a coarser version of a latent continuous variable  $\tilde{Y}_{ij}$ . The observable categorical outcome  $Y_{ij}$  and the unobservable latent variable  $\tilde{Y}_{ij}$  are connected by:

$$Y_{ij} = r_{ij} \quad \Leftrightarrow \quad \theta_{j,r_{ij}-1} < \tilde{Y}_{ij} \leq \theta_{j,r_{ij}}, \quad r_{ij} \in \{1, \dots, K_j\},$$

where  $r_{ij}$  is a category out of  $K_j$  ordered categories and  $\boldsymbol{\theta}_j$  is a vector of suitable threshold parameters for outcome  $j$  with the following restriction:  $-\infty < \theta_{j,1} < \dots < \theta_{j,K_j-1} < \infty$ . Note that in this setting binary observations can be treated as ordinal observations with two categories ( $K_j = 2$ ).

The following linear model is assumed for the relationship between the latent variable  $\tilde{Y}_{ij}$  and the vector of covariates  $\mathbf{x}_{ij}$ :

$$\tilde{Y}_{ij} = \beta_{j0} + \mathbf{x}_{ij}^\top \boldsymbol{\beta}_j + \epsilon_{ij}, \quad (1)$$

where  $\beta_{j0}$  is an intercept term,  $\boldsymbol{\beta}_j = (\beta_{j1}, \dots, \beta_{jp})^\top$  is a vector of regression coefficients, both corresponding to outcome  $j$ , and  $\epsilon_{ij}$  is a mean zero error term. We further assume the  $n$  subjects to be independent and that the error terms are uncorrelated with the covariates. Note that the number of ordered categories  $K_j$  as well as the threshold parameters  $\boldsymbol{\theta}_j$  and the regression coefficients  $\boldsymbol{\beta}_j$  are allowed to vary across outcome dimensions  $j \in J$  to account for possible heterogeneity across the response variables.

**Category-specific regression coefficients** By employing one set of regression coefficients  $\boldsymbol{\beta}_j$  for all categories of the  $j$ -th outcome it is implied that the relationship between the covariates and the responses does not depend on the category. This assumption is called parallel regression or proportional odds assumption (McCullagh 1980) and can be relaxed for one or more covariates by allowing the corresponding regression coefficients to be category specific (see e.g., Peterson and Harrell 1990).

**Link functions** The dependence among the different responses is accounted for by assuming that, for each subject  $i$ , the vector of error terms  $\boldsymbol{\epsilon}_i = [\epsilon_{ij}]_{j \in J_i}$  follows a suitable multivariate distribution. We consider two multivariate distributions which correspond to the

multivariate probit and logit link functions. For the *multivariate probit link* we assume that the errors follow a multivariate normal distribution:  $\epsilon_i \sim \mathcal{N}_{q_i}(\mathbf{0}, \mathbf{\Sigma}_i)$ . A *multivariate logit link* is constructed by employing a multivariate logistic distribution family with univariate logistic margins and a  $t$  copula with certain degrees of freedom. For a vector  $\mathbf{z} = (z_1, \dots, z_q)^\top$ , the multivariate logistic distribution function with  $\nu$  degrees of freedom, mean  $\boldsymbol{\mu}$  and covariance matrix  $\mathbf{\Sigma}$  is defined as:

$$F_{\nu, \boldsymbol{\mu}, \mathbf{\Sigma}}(\mathbf{z}) = t_{\nu, \mathbf{R}}(\{g_\nu((z_1 - \mu_1)/\sigma_1), \dots, g_\nu((z_q - \mu_q)/\sigma_q)\}^\top), \quad (2)$$

where  $t_{\nu, \mathbf{R}}$  is the  $q$  dimensional multivariate  $t$  distribution with  $\nu$  degrees of freedom and correlation matrix  $\mathbf{R}$  corresponding to  $\mathbf{\Sigma}$ ,  $g_\nu(x) = t_\nu^{-1}(\exp(x)/(\exp(x)+1))$ ,  $t_\nu^{-1}$  is the quantile function of the univariate  $t$  distribution with  $\nu$  degrees of freedom and  $\sigma_1^2, \dots, \sigma_q^2$  are the diagonal elements of  $\mathbf{\Sigma}$ .

This  $t$  copula based multivariate logistic family was proposed by O'Brien and Dunson (2004) and later also employed by Nooraee, Abegaz, Ormel, Wit, and van den Heuvel (2016) and Hirk, Hornik, and Vana (2017a), who approximate the distribution by a multivariate  $t$  distribution with the degrees of freedom chosen appropriately. The employed distribution family differs from the conventional multivariate logistic distributions of Gumbel (1961) or Malik and Abraham (1973) in that it offers a more flexible dependence structure through the correlation matrix of the  $t$  copula, while still keeping the log odds interpretation of the regression coefficients through the univariate logistic margins.

## 2.2. Identifiability issues

As the absolute scale and the absolute location are not identifiable in ordinal models, further restrictions on the parameter set need to be imposed. Assuming  $\mathbf{\Sigma}_i$  to be a covariance matrix with diagonal elements  $[\sigma_{ij}^2]_{j \in J_i}$ , only the quantities  $\beta_j/\sigma_{ij}$  and  $(\theta_{j,ri} - \beta_{j0})/\sigma_{ij}$  are identifiable in the model in Equation 1. Hence, in order to obtain an identifiable model the parameter set is typically constrained in one of the following ways:

- Fixing the intercept  $\beta_{j0}$  (e.g., to zero), using flexible thresholds  $\theta_j$  and fixing  $\sigma_{ij}$  (e.g., to unity)  $\forall j \in J_i, \forall i \in \{1, \dots, n\}$ ;
- Leaving the intercept  $\beta_{j0}$  unrestricted, fixing one threshold parameter (e.g.,  $\theta_{j,1} = 0$ ) and fixing  $\sigma_{ij}$  (e.g., to unity)  $\forall j \in J_i, \forall i \in \{1, \dots, n\}$ ;
- Fixing the intercept  $\beta_{j0}$  (e.g., to zero), fixing one threshold parameter (e.g.,  $\theta_{j,1} = 0$ ) and leaving  $\sigma_{ij}$  unrestricted  $\forall j \in J_i, \forall i \in \{1, \dots, n\}$ ;
- Leaving the intercept  $\beta_{j0}$  unrestricted, fixing two threshold parameters (e.g.,  $\theta_{j,1} = 0$  and  $\theta_{j,2} = 1$ ) and leaving  $\sigma_{ij}$  unrestricted  $\forall j \in J_i, \forall i \in \{1, \dots, n\}$ .<sup>1</sup>

Note that the first two options are the most commonly used in the literature. All of these alternative model parameterizations are supported by the **mvord** package, allowing the user to choose the most convenient one for each specific application. Table 1 in Section 3.5 gives an overview on the identifiable parameterizations implemented in the package.

<sup>1</sup>Note that this parameterization cannot be applied to the binary case.

### 2.3. Error structures

#### Basic model

The *basic* multivariate ordinal regression model assumes that the correlation (and possibly variance, depending on the parameterization) parameters in the distribution function of the  $\epsilon_i$ 's are constant for all subjects  $i$ .

**Correlation** The dependence between the multiple measurements or outcomes can be captured by different correlation structures. Among them, we concentrate on the following three:

- the *general* correlation structure assumes different correlation parameters between pairs of outcomes  $\text{corr}(\epsilon_{ik}, \epsilon_{il}) = \rho_{kl}$ ;
- the *equicorrelation* structure  $\text{corr}(\epsilon_{ik}, \epsilon_{il}) = \rho$  implies that the correlation between all pairs of outcomes is constant;
- when faced with longitudinal data, especially when moderate to long subject-specific time series are available, an  $AR(1)$  *autoregressive* correlation model of order one can be employed. For given equally spaced time points  $t_1, t_2, \dots, t_q$  this  $AR(1)$  error structure implies an exponential decay in the correlation with the lag. If  $t_k$  and  $t_l$  are the time points when  $Y_{ik}$  and  $Y_{il}$  are observed, then  $\text{corr}(\epsilon_{ik}, \epsilon_{il}) = \rho^{|t_k - t_l|}$ .

**Variance** If a parameterization which supports the estimation of the variance of the latent processes is used (see Section 2.2), it is assumed that  $\text{VAR}(\epsilon_{ij}) = \sigma_j^2$ .

#### Extending the basic model

In some applications, the constant correlation (and variance) structure across subjects may be too restrictive. We hence extend the basic model by allowing the use of covariates in the correlation (and variance) specifications.

**Correlation** For each subject  $i$  and each pair  $(k, l)$  from the set  $J_i$ , the correlation parameter  $\rho_{ikl}$  is assumed to depend on a vector  $\mathbf{s}_i$  of  $m$  subject-specific covariates. The hyperbolic tangent transformation allows us to reparameterize the linear term  $\alpha_{0kl} + \mathbf{s}_i^\top \boldsymbol{\alpha}_{kl}$  in terms of a correlation parameter:

$$\frac{1}{2} \log \left( \frac{1 + \rho_{ikl}}{1 - \rho_{ikl}} \right) = \alpha_{0kl} + \mathbf{s}_i^\top \boldsymbol{\alpha}_{kl}, \quad \rho_{ikl} = \frac{e^{2(\alpha_{0kl} + \mathbf{s}_i^\top \boldsymbol{\alpha}_{kl})} - 1}{e^{2(\alpha_{0kl} + \mathbf{s}_i^\top \boldsymbol{\alpha}_{kl})} + 1}.$$

If  $\boldsymbol{\alpha}_{kl} = \mathbf{0}$  for all  $k, l \in J_i$ , this model would correspond to the general correlation structure in the basic model. Moreover, if  $\alpha_{0kl} = 0$  and  $\boldsymbol{\alpha}_{kl} = \mathbf{0}$  for all  $k, l \in J_i$ , the correlation matrix is the identity matrix and the responses are uncorrelated.

For the more parsimonious error structures of equicorrelation and  $AR(1)$ , in the extended model the correlation parameters are modeled as:

$$\frac{1}{2} \log \left( \frac{1 + \rho_i}{1 - \rho_i} \right) = \alpha_0 + \mathbf{s}_i^\top \boldsymbol{\alpha}, \quad \rho_i = \frac{e^{2(\alpha_0 + \mathbf{s}_i^\top \boldsymbol{\alpha})} - 1}{e^{2(\alpha_0 + \mathbf{s}_i^\top \boldsymbol{\alpha})} + 1}.$$

**Variance** Similarly, one could model the heterogeneity among the subjects through the variance parameters  $\text{VAR}(\epsilon_{ij}) = \sigma_{ij}^2$  by employing the following linear model on the log-variance:

$$\log(\sigma_{ij}^2) = \gamma_{0j} + \mathbf{s}_i^\top \boldsymbol{\gamma}_j.$$

The correlation (or covariance) matrix  $\boldsymbol{\Sigma}_i$  must be positive-semi-definite. This can be ensured by the use of special algorithms such as the one proposed by [Higham \(1988\)](#).

## 2.4. Composite likelihood estimation

In order to estimate the model parameters we use a composite likelihood approach, where the full likelihood is approximated by a pseudo-likelihood which is constructed from lower dimensional marginal distributions, more specifically by “aggregating” the likelihoods corresponding to pairs of observations ([Varin, Reid, and Firth 2011](#)).

For a given parameter vector  $\boldsymbol{\delta}$ , which contains the threshold parameters, the regression coefficients and the parameters of the error structure, the likelihood is given by:

$$\mathcal{L}(\boldsymbol{\delta}) = \prod_{i=1}^n \mathbb{P} \left( \bigcap_{j \in J_i} \{Y_{ij} = r_{ij}\} \right)^{w_i} = \prod_{i=1}^n \left( \int_{D_i} f_{q_i}(\tilde{\mathbf{Y}}_i; \boldsymbol{\delta}) d^{q_i} \tilde{\mathbf{Y}}_i \right)^{w_i},$$

where  $D_i = \prod_{j \in J_i} (\theta_{j,r_{ij}-1}, \theta_{j,r_{ij}}]$  is a Cartesian product,  $w_i$  are subject-specific non-negative weights, which are set to one in the default case, and  $f_{q_i}$  is the  $q_i$ -dimensional density of the error terms  $\boldsymbol{\epsilon}_i$ . We approximate this full likelihood by a pairwise likelihood which is constructed from bivariate marginal distributions. If the number of observed outcomes for subject  $i$  is less than two ( $q_i < 2$ ), the univariate marginal distribution enters the likelihood. The pairwise log-likelihood function is obtained by:

$$p\ell(\boldsymbol{\delta}) = \sum_{i=1}^n w_i \left[ \mathbb{1}_{\{q_i \geq 2\}} \sum_{\substack{k < l \\ k, l \in J_i}} \log(\mathbb{P}(Y_{ik} = r_{ik}, Y_{il} = r_{il})) + \right. \\ \left. \mathbb{1}_{\{q_i = 1\}} \mathbb{1}_{\{k \in J_i\}} \log(\mathbb{P}(Y_{ik} = r_{ik})) \right]. \quad (3)$$

Denoting by  $U_{ij} = (\theta_{j,r_{ij}} - \beta_{j0} - \mathbf{x}_{ij}^\top \boldsymbol{\beta}_j) / \sigma_{ij}$  the upper and by  $L_{ij} = (\theta_{j,r_{ij}-1} - \beta_{j0} - \mathbf{x}_{ij}^\top \boldsymbol{\beta}_j) / \sigma_{ij}$  the lower integration bounds and by  $f_1$  and  $f_2$  the uni- and bivariate density functions corresponding to the error distribution, the uni- and bivariate probabilities are given by:

$$\mathbb{P}(Y_{ik} = r_{ik}, Y_{il} = r_{il}) = \int_{L_{ik}}^{U_{ik}} \int_{L_{il}}^{U_{il}} f_2(v_{ik}, v_{il}; \rho_{ikl}) dv_{ik} dv_{il}, \\ \mathbb{P}(Y_{ik} = r_{ik}) = \int_{L_{ik}}^{U_{ik}} f_1(v_{ik}) dv_{ik}.$$

The maximum pairwise likelihood estimates  $\hat{\boldsymbol{\delta}}_{\text{PL}}$  are obtained by direct maximization of the composite likelihood given in Equation 3. The threshold and error structure parameters to be estimated are reparameterized such that unconstrained optimization can be performed. Firstly, we reparameterize the threshold parameters in order to achieve monotonicity. Secondly, for all unrestricted correlation (and covariance) matrices we use the spherical parameterization of [Pinheiro and Bates \(1996\)](#). This parameterization has the advantage that it



can be easily applied to correlation matrices. Thirdly, if we assume to have equicorrelated or  $AR(1)$  errors, we use the hyperbolic tangent transformation.

Computation of the standard errors is needed in order to quantify the uncertainty of the maximum pairwise likelihood estimates. Under certain regularity conditions, the maximum pairwise likelihood estimates are consistent as the number of responses is fixed and  $n \rightarrow \infty$ . In addition, the maximum pairwise likelihood estimator is asymptotically normal with asymptotic mean  $\boldsymbol{\delta}$  and a covariance matrix which equals the inverse of the Godambe information matrix:

$$G(\boldsymbol{\delta})^{-1} = H^{-1}(\boldsymbol{\delta})V(\boldsymbol{\delta})H^{-1}(\boldsymbol{\delta}),$$

where  $H(\boldsymbol{\delta})$  the Hessian (sensitivity matrix) and  $V(\boldsymbol{\delta})$  the variability matrix. The variability matrix  $V(\boldsymbol{\delta})$  and the Hessian  $H(\boldsymbol{\delta})$  can be estimated as follows:

$$\begin{aligned}\hat{V}(\boldsymbol{\delta}) &= \frac{1}{n} \sum_{i=1}^n \left( \frac{\partial p\ell_i(\hat{\boldsymbol{\delta}}_{\text{PL}})}{\partial \boldsymbol{\delta}} \right) \left( \frac{\partial p\ell_i(\hat{\boldsymbol{\delta}}_{\text{PL}})}{\partial \boldsymbol{\delta}} \right)^\top, \\ \hat{H}(\boldsymbol{\delta}) &= -\frac{1}{n} \sum_{i=1}^n \frac{\partial^2 p\ell_i(\hat{\boldsymbol{\delta}}_{\text{PL}})}{\partial \boldsymbol{\delta} \partial \boldsymbol{\delta}^\top} \\ &= \frac{1}{n} \sum_{i=1}^n \sum_{\substack{k < l \\ k, l \in J_i}} \left( \frac{\partial p\ell_{ikl}(\hat{\boldsymbol{\delta}}_{\text{PL}})}{\partial \boldsymbol{\delta}} \right) \left( \frac{\partial p\ell_{ikl}(\hat{\boldsymbol{\delta}}_{\text{PL}})}{\partial \boldsymbol{\delta}} \right)^\top,\end{aligned}$$

where  $p\ell_i(\boldsymbol{\delta})$  is the component of the pairwise log-likelihood corresponding to subject  $i$  and  $p\ell_{ikl}(\boldsymbol{\delta})$  corresponds to subject  $i$  and pair  $(k, l)$ .

In order to compare different models, the composite likelihood information criterion by [Varin and Vidoni \(2005\)](#) can be used:  $\text{CLIC}(\boldsymbol{\delta}) = -2 \, p\ell(\hat{\boldsymbol{\delta}}_{\text{PL}}) + k \, \text{tr}(\hat{V}(\boldsymbol{\delta})\hat{H}(\boldsymbol{\delta})^{-1})$  (where  $k = 2$  corresponds to CLAIC and  $k = \log(n)$  corresponds to CLBIC). A comprehensive overview and further details on the properties of the maximum composite likelihood estimates is provided in [Varin \(2008\)](#).

## 2.5. Interpretation of the coefficients

Unlike in linear regression models, the interpretation of the regression coefficients and of the threshold parameters in ordinal models is not straightforward. Estimated thresholds and coefficients represent only signal to noise ratios and cannot be interpreted directly (see [Section 2.2](#)). For one particular outcome  $j$ , the coefficients can be interpreted in the same way as in univariate cumulative link models. Let us assume without loss of generality that a higher latent score leads to better ratings on the ordinal scale. This implies that the first category is the worst and category  $K_j$  is the best category. The marginal cumulative probabilities implied by the model in [Equation 1](#) are then given by the following relationship:

$$P(Y_{ij} \leq r_{ij} | \mathbf{x}_{ij}) = P(\mathbf{x}_{ij}^\top \boldsymbol{\beta}_j + \epsilon_{ij} \leq \theta_{j,r_{ij}}) = P(\epsilon_{ij} \leq \theta_{j,r_{ij}} - \mathbf{x}_{ij}^\top \boldsymbol{\beta}_j) = F_1(\theta_{j,r_{ij}} - \mathbf{x}_{ij}^\top \boldsymbol{\beta}_j),$$

where  $F_1$  is the marginal (univariate) distribution function of the errors for the  $j$ -th outcome.



One natural way to interpret ordinal regression models is to analyze partial effects, where one is interested in how a marginal change in one variable  $x_{ijv}$  changes the outcome distribution. The partial probability effects in the cumulative model are given by:

$$\delta_{r,v}^j(\mathbf{x}_{ij}) = \frac{\partial \mathbb{P}(Y_{ij} = r_{ij} | \mathbf{x}_{ij})}{\partial x_{ijv}} = - \left( f_1(\theta_{j,r_{ij}} - \mathbf{x}_{ij}^\top \boldsymbol{\beta}_j) - f_1(\theta_{j,r_{ij}-1} - \mathbf{x}_{ij}^\top \boldsymbol{\beta}_j) \right) \beta_{jv},$$

where  $f_1$  is the density corresponding to  $F_1$ ,  $x_{ijv}$  is the  $v$ -th element in  $\mathbf{x}_{ij}$  and  $\beta_{jv}$  is the  $v$ -th element in  $\boldsymbol{\beta}_j$ . In case of discrete variables it is more appropriate to consider the changes in probability before and after the change in the variable instead of the partial effects using:

$$\Delta \mathbb{P}(Y_{ij} = r_{ij} | \mathbf{x}_{ij}, \tilde{\mathbf{x}}_{ij}) = \mathbb{P}(Y_{ij} = r_{ij} | \tilde{\mathbf{x}}_{ij}) - \mathbb{P}(Y_{ij} = r_{ij} | \mathbf{x}_{ij}),$$

where all elements of  $\tilde{\mathbf{x}}_{ij}$  are equal to  $\mathbf{x}_{ij}$  except for the  $v$ -th element, which is equal to  $\tilde{x}_{ijv} = x_{ijv} + \Delta x_{ijv}$  for the discrete change  $\Delta x_{ijv}$  in the variable  $x_v$ . We refer to [Greene and Hensher \(2010\)](#) and [Boes and Winkelmann \(2006\)](#) for further discussion of the interpretation of partial effects in ordered response models.

In the presence of the probit link function, we have the following relationship between the cumulative probabilities and the latent process:

$$\Phi^{-1}(\mathbb{P}(Y_{ij} \leq r_{ij} | \mathbf{x}_{ij})) = \theta_{j,r_{ij}} - \mathbf{x}_{ij}^\top \boldsymbol{\beta}_j.$$

An increase of one unit in a variable  $x_{jv}$  (given that all other variables are held constant) changes the probit of the probability that category  $r$  or lower is observed by the value of the coefficient  $\beta_{jv}$  of this variable. In other words  $\mathbb{P}(Y_{ij} \leq r_{ij} | \mathbf{x}_{ij})$ , the probability that category  $r_{ij}$  or lower is observed, changes by the increase/decrease in the distribution function. Moreover, predicted probabilities for all ordered response categories can be calculated and compared for given sets of explanatory variables.

In the presence of the logit link function, the regression coefficients of the underlying latent process are scaled in terms of marginal log odds ([McCullagh 1980](#)):

$$\log \left( \frac{\mathbb{P}(Y_{ij} \leq r_{ij} | \mathbf{x}_{ij})}{\mathbb{P}(Y_{ij} > r_{ij} | \mathbf{x}_{ij})} \right) = \theta_{j,r_{ij}} - \mathbf{x}_{ij}^\top \boldsymbol{\beta}_j.$$

For a one unit increase in one variable  $x_{jv}$  holding all the others constant, we expect a change of size of the coefficient  $\beta_{jv}$  of this variable in the expected value on the log odds scale. Due to the fact that the marginal effects of the odds ratios do not depend on the category, one often exponentiates the coefficients in order to obtain the following convenient interpretation in terms of odds ratios:

$$\frac{\mathbb{P}(Y_{ij} \leq r_{ij} | \mathbf{x}_{ij}) / \mathbb{P}(Y_{ij} > r_{ij} | \mathbf{x}_{ij})}{\mathbb{P}(Y_{ij} \leq r_{ij} | \tilde{\mathbf{x}}_{ij}) / \mathbb{P}(Y_{ij} > r_{ij} | \tilde{\mathbf{x}}_{ij})} = \exp((\tilde{\mathbf{x}}_{ij} - \mathbf{x}_{ij})^\top \boldsymbol{\beta}_j).$$

This means for a one unit increase in  $x_{jv}$  holding all the other variables constant, changes the odds ratio by  $e^{\beta_{jv}}$ . In other words, the odds after a one unit change in  $x_{jv}$  are the odds before the change multiplied by  $e^{-\beta_{jv}}$

$$\frac{\mathbb{P}(Y_{ij} \leq r_{ij} | \mathbf{x}_{ij})}{\mathbb{P}(Y_{ij} > r_{ij} | \mathbf{x}_{ij})} \exp(-\beta_j) = \frac{\mathbb{P}(Y_{ij} \leq r_{ij} | \tilde{\mathbf{x}}_{ij})}{\mathbb{P}(Y_{ij} > r_{ij} | \tilde{\mathbf{x}}_{ij})}.$$

If the regression coefficients vary across the multiple responses, they cannot be compared directly due to the fact that the measurement units of the underlying latent processes differ. Nevertheless, one possibility to compare coefficients is through concept of *importance*. Reusens and Croux (2017) extend an approach for comparing coefficients of probit and logit models by Hoetker (2007) in order to compare the coefficients across repeated measurements. They analyze the *importance ratio*

$$R_{jv} = \frac{\beta_{jv}}{\beta_{j,base}},$$

where  $\beta_{j,base}$  is the coefficient of a *base* variable and  $v$  is one of the remaining  $p - 1$  variables. This ratio can be interpreted as follows: A one unit increase in the variable  $v$  has in expectation the same effect in the *base* variable multiplied by the ratio  $R_{jv}$ . Another interpretation is the so called *compensation variation*: The ratio is the required increase in the *base* variable that is necessary to compensate a one unit decrease in the variable  $v$  in a way that the score of the outcome remains the same.

### 3. Implementation

Multivariate ordinal regression models in the R package **mvord** are fitted using the function `mvord()`:

```
mvord(formula, data, error.structure = cor_general(~ 1), link = mvprobit(),
      index = NULL, response.names = NULL, response.levels = NULL,
      threshold.constraints = NULL, threshold.values = NULL,
      coef.constraints = NULL, coef.values = NULL, weights = NULL, se = TRUE,
      offset = NULL, scale = FALSE, start.values = NULL, solver = "newuoa",
      PL.lag = NULL, control = list(maxit=200000, trace = 1, kkt = FALSE))
```

#### 3.1. Data structure

We use a long format for the input of `data`, where each row contains a subject index  $i$  (`subject_index`), a multiple measurement index  $j$  (`multiple_measurement_index`), an ordinal response ( $Y$ ) and all the covariates ( $X_1$  to  $X_p$ )<sup>2</sup>. This long format data structure is internally transformed to an  $n \times q$  matrix of responses  $Y_1$  to  $Y_q$  (which contains NA in the case of missing entries) and a list of covariate matrices  $X_j$  for all  $j \in J$ . In order to construct these objects, the subject index  $i$  and the multiple measurement index  $j$  should be specified. This can be performed by an optional argument `index`, a character vector of length two, specifying the column names of the subject index and the multiple measurement index in `data`.

```
index = c("subject_index", "multiple_measurement_index")
```

The default value of `index` is `NULL` assuming that the first column of `data` contains the subject index  $i$  and the second column the multiple measurement index  $j$ .

---

<sup>2</sup>If the covariates have different scales, numerical instabilities may occur in the estimation procedure. In such cases we suggest to e.g., standardize the covariates or transform them to have a similar scale.

If specific constraints are imposed on the threshold parameters and/or on the regression coefficients, it is important to know which level of the multiple measurement index  $j$  corresponds to the first dimension, second dimension and so on. Hence, a well defined index  $j \in J$  for the multiple measurements is needed. Therefore, a vector `response.names` is used to define the index number of the multiple measurement. If we assume to have  $q = 4$  outcomes `Y1` to `Y4`, the following `response.names` are set:

```
response.names = c("Y1", "Y2", "Y3", "Y4")
```

The default value of `response.names` is `NULL` giving the natural ordering of the levels of the factor variable for all the multiple measurements. The ordering of `response.names` always specifies the index of the multiple measurement unit  $j \in J$ . This ordering is essential when putting constraints on the parameters and when setting `response.levels`.

```
response.levels = list(Y1 = c("F", "E", "D", "C", "B", "A"),
                      Y2 = c("F", "E", "D", "C", "B", "A"),
                      Y3 = c("M", "L", "K", "J", "I", "H", "G"),
                      Y4 = c("O", "N"))
```

If the categories differ across multiple measurements (either the number of categories and/or the category labels) one needs to specify the `response.levels` explicitly. This is performed by a list of length  $q$ , where each element contains the names of the levels of the ordered categories in ascending (or if desired descending) order.

### 3.2. Formula

The ordinal responses `Y` and the  $p$  covariates `X1`, `X2`, ..., `Xp` are passed by a `formula` object. Intercepts can be included or excluded in the model depending on the model parameterization chosen in order to ensure identifiability:

**Model without intercept** If the intercept should be removed, the `formula` has the following form:

```
formula = Y ~ 0 + X1 + ... + Xp
```

**Model with intercept** If one wants to include an intercept in the model, there are two equivalent possibilities to set the model `formula`. Either the intercept is included explicitly by:

```
formula = Y ~ 1 + X1 + ... + Xp
```

or by

```
formula = Y ~ X1 + ... + Xp
```

**Note on intercept in formula** We differ in our approach of specifying the model `formula` from the formula objects in e.g., `MASS::polr()` or `ordinal::clm()`, in that we allow the user to specify models without intercept. This option is not supported in the **MASS** and **ordinal** packages, where an intercept is always specified in `formula` as the threshold parameters are treated as intercepts. We choose to allow for this option, in order to have a correspondence to the identifiability constraints presented in Section 2.2.

Even so, the user should be aware that the threshold parameters are basically category and outcome specific intercepts. This implies that, even if the intercept is explicitly removed from the model through the `formula` object and hence set to zero, the rest of the covariates should be specified in such a way that multicollinearity does not arise (as if an intercept were specified). This is of primary importance when including categorical covariates, where one category will be taken as baseline by default.

### 3.3. Link functions

The multivariate link functions are specified as objects of class `'mvlink'`. An object of class `'mvlink'` is a list with elements specifying the distribution function of the errors, functions for computing the corresponding univariate and bivariate probabilities, as well as additional arguments specific to each link. If gradient functions passed on, these will be used in the computation of the standard errors. This design has been inspired by the design of the `'family'` class in package **stats** and it facilitates the addition of new link functions to the package.

We offer two different multivariate link functions, the multivariate probit link and a multivariate logit link. For the multivariate probit link a multivariate normal distribution for the errors is applied. The normal bivariate probabilities which enter the pairwise log-likelihood are computed with package **pbivnorm** (Genz and Kenkel 2015). The multivariate probit link can be specified by:

```
link = mvprobit()
```

For the multivariate logit link a  $t$  copula based multivariate distribution with logistic margins is used (as explained in Section 2.1) and can be specified by:

```
link = mvlogit(df = 8L)
```

The `mvlogit()` function has an optional integer valued argument `df` which specifies the degrees of freedom to be used for the  $t$  copula. The default value of the degrees of freedom parameter is 8. When choosing  $\nu \approx 8$ , the multivariate logistic distribution in Equation 2 is well approximated by a multivariate  $t$  distribution (O'Brien and Dunson 2004). This is also the value chosen by Noorae *et al.* (2016) in their analysis. We restrict the degrees of freedom to be integer valued because the most efficient routines for computing bivariate  $t$  probabilities do not support non-integer degrees of freedom. We use the Fortran code from Alan Genz (Genz and Bretz 2009) to compute the bivariate  $t$  probabilities. As the degrees of freedom parameter is integer valued, we do not estimate it in the optimization procedure. If the optimal degrees of freedom are of interest, we leave the task of choosing an appropriate grid of values of `df` to the user, who should then estimate a separate model for each value in the grid. The best model can be chosen by CLAIC or CLBIC.

### 3.4. Error structures

Different error structures are implemented in **mvord** and can be specified through the argument `error.structure`. The error structure objects are of class `'error_struct'`. This approach is slightly different than in package **nlme**, where the error structure is defined by two classes: `'varFunc'` for the variance function and `'corStruct'` for the correlation structure. We also define the following subclasses for the error structures: `'cor_general'` (similar to **nlme**'s `'corSymm'`), `'cor_equi'` (similar to `'corCompSymm'`), `'cor_ar1'` (similar to `'corAR1'`) and `'cov_general'` (similar to `'corSymm'` with variance function `'varIdent'`). The different error structures are chosen in the following way:

#### *Basic model*

**Correlation** For the basic model specification the following correlation structures are implemented in **mvord**:

- `cor_general()` – A general error structure, where the correlation matrix of the error terms is unrestricted and constant across all subjects:  $\text{corr}(\epsilon_{ik}, \epsilon_{il}) = \rho_{kl}$ . This error structure is among the most common in the literature (e.g., [Scott and Kanaroglou 2002](#); [Bhat, Varin, and Ferdous 2010](#); [Kenne Pagui and Canale 2016](#)).

```
error.structure = cor_general(formula = ~ 1)
```

- `cor_equi()` – An equicorrelation structure with  $\text{corr}(\epsilon_{ik}, \epsilon_{il}) = \rho$  is used.

```
error.structure = cor_equi(formula = ~ 1)
```

- `cor_ar1()` – An autoregressive error structure of order one with  $\text{corr}(\epsilon_{ik}, \epsilon_{il}) = \rho^{|t_k - t_l|}$  is used.

```
error.structure = cor_ar1(formula = ~ 1)
```

**Variance** A model with variance parameters  $\text{VAR}(\epsilon_{ij}) = \sigma_j^2$  corresponding to each outcome, when the identifiability requirements are fulfilled, can be specified in the following way:

- the estimation of  $\sigma_j^2$  is only implemented in combination with the general correlation structure. Hence, the unrestricted covariance matrix of the error terms can be estimated by:

```
error.structure = cov_general(formula = ~ 1)
```

#### *Extending the basic model*

## Correlation

- For the heterogeneous general correlation structure, the current implementation only allows the use of one `factor` variable `f` as covariate. As previously mentioned, this factor variable should be subject-specific and hence should not vary across the multiple responses. This implies that a correlation matrix will be estimated for each factor level.

```
error.structure = cor_general(formula = ~ f)
```

- Estimating an equicorrelation structure depending on subject-specific covariates:

```
error.structure = cor_equi(formula = ~ S1 + ... + Sm)
```

- Estimating an  $AR(1)$  correlation structure depending on subject-specific covariates:

```
error.structure = cor_ar1(formula = ~ S1 + ... + Sm)
```

## Variance

- As in the basic model, the estimation of the heterogeneous variance parameters can be performed for the general correlation structure. A subject-specific factor `f` can be used as covariate in the log variance equation. This can be specified by:

```
error.structure = cov_general(formula = ~ f)
```

In addition to the correlation matrices, which are estimated for each factor level of `f`, a vector of dimension  $q$  of variance parameters will be estimated for each factor level.

### 3.5. Constraints on thresholds

The package supports constraints on the threshold parameters. Firstly, the user can specify whether the threshold parameters should be equal across some or all response dimensions. Secondly, the values of some of the threshold parameters can be fixed. This feature is important for the users who wish to further restrict the parameter space of the thresholds or who wish to specify values for the threshold parameters other than the default values used in the package. Note that fixing some of the thresholds is needed for some of the parameterizations presented in Table 1 in order to ensure identifiability of the model.

#### *Threshold constraints across responses*

Such constraints can be imposed by a vector of positive integers `threshold.constraints`, where dimensions with equal threshold parameters get the same integer. When restricting two outcome dimensions to be equal, one has to be careful that the number of categories in the two outcome dimensions must be the same. In an example with  $q = 4$  different outcomes, if one wishes to restrict the threshold parameters of the first and second outcomes `Y1` and `Y2` to be equal ( $\theta_1 = \theta_2$ ). These restrictions can be specified as:

```
threshold.constraints = c(1, 1, 2, 3)
```

### Fixing threshold values

Values for the threshold parameters can be specified by the argument `threshold.values`. For this purpose the user can pass a `list` with  $q$  elements, where each element is a `vector` of length  $K_j - 1$  (where  $K_j$  is number of ordered categories for ordinal outcome  $j$ ). A numeric value in this vector fixes the corresponding threshold parameter to the specified value while `NA` leaves the parameter flexible and indicates it should be estimated.

After specifying the error structure (through the `error.structure` argument) and whether an intercept should be estimated or not (in the `formula` argument), the user can choose among five possible options for fixing the thresholds:

- leaving all thresholds flexible;
- fixing the first threshold  $\theta_{j,1}$  to a constant  $a_j$  for all  $j \in J$ ;
- fixing the first and second thresholds  $\theta_{j,1} = a_j$ ,  $\theta_{j,2} = b_j$  for all outcomes with  $K_j > 2$ ;
- fixing the first and last thresholds  $\theta_{j,1} = a_j$ ,  $\theta_{j,K_j-1} = b_j$  for all outcomes with  $K_j > 2$ ;
- an extra option is fixing all of the threshold parameters, for all  $j \in J$ .

Note that the option chosen needs to be consistent across the different outcomes (e.g., it is not allowed to fix the first and the last threshold for one outcome and the first and the second threshold for a different). Table 1 provides information about the options available for each combination error structure and intercept, as well as about the default values in case the user does not specify any threshold values.

Error structure	Intercept	Threshold parameters				
		all flexible	one fixed $\theta_{j,1} = a_j$	two fixed $\theta_{j,1} = a_j$ $\theta_{j,2} = b_j$	two fixed $\theta_{j,1} = a_j$ $\theta_{j,K_j-1} = b_j$	all fixed
<b>cor</b>	no	✓	✓	✓	✓	✓
	yes		✓	✓	✓	✓
<b>cov</b>	no		✓	✓	✓	✓
	yes			✓	✓	✓

Table 1: This table displays different model parameterizations in the presence of ordinal observations ( $K_j > 2 \forall j \in J$ ). The row **cor** includes error structures **cor\_general**, **cor\_equi** and **cor\_ar1**, while row **cov** includes the error structure **cov\_general**. The minimal restrictions (default) to ensure identifiability are given in green. The default threshold values (in case `threshold.values = NULL`) are always  $a_j = 0$  and  $b_j = 1$ .

In the presence of binary observations ( $K_j = 2$ ), if a **cov\_general** error structure is used, the intercept has always to be fixed to some value due to identifiability constraints. In a correlation structure setting no further restrictions are required.

For example, the following restrictions on the threshold parameters:

- $\theta_{11} = -4 \leq \theta_{12} \leq \theta_{13} \leq \theta_{14} \leq \theta_{15} \leq \theta_{16}$ ;
- $\theta_{21} = -4 \leq \theta_{22} \leq \theta_{23} \leq \theta_{24} \leq \theta_{25} \leq \theta_{26}$ ;



- $\theta_{31} = -5 \leq \theta_{32} \leq \theta_{33} \leq \theta_{34} \leq \theta_{35} \leq \theta_{36} \leq \theta_{37}$ ;
- $\theta_{41} = 0$ ,

are implemented as:

```
threshold.values = list(c(-4, NA, NA, NA, NA, NA), c(-4, NA, NA, NA, NA, NA),
  c(-5, NA, NA, NA, NA, NA), c(0))
```

### 3.6. Constraints on coefficients

The package supports constraints on the regression coefficients. Firstly, the user can specify whether the regression coefficients should be equal across some or all response dimensions. Secondly, the values of some of the regression coefficients can be fixed.

As there is no unanimous way to specify such constraints, we offer two options. The first option is similar to the specification of constraints on the thresholds. The constraints can be specified in this case as a vector or matrix of integers, where coefficients getting same integer value are set equal. Values of the regression coefficients can be fixed through a matrix. Alternatively constraints on the regression coefficients can be specified by using the design employed by the **VGAM** package. The constraints in this setting are set through a named list, where each element of the list contains a matrix full-column rank. If the values of some regression coefficients should be fixed, offsets can be used. This design has the advantage that it supports constraints on outcome-specific as well as category-specific regression coefficients. While the first option has the advantage of requiring a more concise input, it does not support category-specific coefficients. The second option offers a more flexible design in this respect.

#### *Coefficient constraints across responses*

Such constraints can be specified by the argument `coef.constraints`, which can be either a vector or a matrix of integer values. If vector constraints of the type  $\beta_k = \beta_l$ , are desired, which should hold for all  $p$  regression coefficients corresponding to outcome  $k$  and  $l$ , the easiest way to specify this is by means of a vector of integers of dimension  $q$ , where outcomes with equal vectors of regression coefficients get the same integer. For example, for  $q = 4$ , a model where the regression coefficients of the first and second outcomes are equal ( $\beta_1 = \beta_2$ ), while the coefficients of outcomes three and four are unrestricted, can be specified as:

```
coef.constraints = c(1, 1, 2, 3)
```

A more flexible framework allows the user to specify such constraints for each of the regression coefficients of the  $p$  covariates, not only for the whole vector. Such constraints will be specified by means of a matrix of dimension  $q \times p$ , where each column specifies constraints for one of the  $p$  covariates in the same way as presented above. Moreover, a value of `NA` indicates that the corresponding coefficient is fixed (as we will show below) and should not be estimated.

Let us assume we are dealing with seven covariates and four outcomes (as in the example presented in Section 5.1) and the following constraints on the regression coefficients:

- $\beta_{12} = \beta_{22} = \beta_{32}$ ;

- $\beta_{13} = 0, \beta_{23} = 0, \beta_{33} = 0;$
- $\beta_{14} = \beta_{24} = \beta_{34}, \beta_{44} = 0;$
- $\beta_{15} = \beta_{25} = \beta_{35} = \beta_{45} = 2.$

These restrictions on the regression coefficients are imposed by:

```
coef.constraints = cbind(c(1, 2, 3, 4), c(1, 1, 1, 2), c(NA, NA, NA, 1),
  c(1, 1, 1, NA), c(NA, NA, NA, NA), c(1, 2, 3, 4), c(1, 2, 3, 4))
```

Specific values of coefficients can be fixed through the `coef.values` argument, as we will show in the following.

#### *Fixing coefficient values*

In addition, specific values on the regression coefficients can be set in the  $q \times p$  matrix `coef.values`. Again each column corresponds to the regression coefficients of one covariate. This feature is to be used if some of the covariates have known slopes, but also for excluding covariates from the mean model of some of the outcomes (by fixing the regression coefficient to zero).

By default, if no `coef.values` are passed by the user, all the regression coefficients which receive an NA in `coef.constraints` will be set to zero. NA in the `coef.values` matrix indicates the regression coefficient ought to be estimated. Setting `coef.values` in accordance with the `coef.constraints` from above:

```
coef.values = cbind(c(NA, NA, NA, NA), c(NA, NA, NA, NA), c(0, 0, 0, NA),
  c(NA, NA, NA, 0), c(2, 2, 2, 2), c(NA, NA, NA, NA), c(NA, NA, NA, NA))
```

gives the following equations for the latent scores:

$$\begin{aligned}
 \tilde{Y}_{i1} &= \beta_{11}x_{i1} + \beta_{12}x_{i2} && + \beta_{14}x_{i4} + 2x_{i5} + \beta_{16}x_{i6} + \beta_{17}x_{i7}, \\
 \tilde{Y}_{i2} &= \beta_{21}x_{i1} + \beta_{12}x_{i2} && + \beta_{14}x_{i4} + 2x_{i5} + \beta_{26}x_{i6} + \beta_{27}x_{i7}, \\
 \tilde{Y}_{i3} &= \beta_{31}x_{i1} + \beta_{12}x_{i2} && + \beta_{14}x_{i4} + 2x_{i5} + \beta_{36}x_{i6} + \beta_{37}x_{i7}, \\
 \tilde{Y}_{i4} &= \beta_{41}x_{i1} + \beta_{42}x_{i2} + \beta_{43}x_{i3} && + 2x_{i5} + \beta_{46}x_{i6} + \beta_{47}x_{i7}.
 \end{aligned}$$

#### *Constraints on category-specific coefficients*

If the parallel regression or proportional odds assumption ought to be relaxed, the constraint design of package **VGAM** can be employed. Let us consider the model from above and further assume outcome one, two and three are ordinal variables with 3 categories and outcome four has two categories. For illustration purposes we now relax the parallel regression assumption partially for covariates X1 and X3 in the following way:

- $\beta_{11, \text{cat}_1} = \beta_{11, \text{cat}_2} \neq \beta_{11, \text{cat}_3};$
- $\beta_{43, \text{cat}_1} \neq \beta_{43, \text{cat}_2},$

where  $\beta_{jk, \text{cat}_r}$  denotes the regression coefficient for multiple measurement  $j$ , covariate  $k$  and category  $r$ . By the first restriction, for the first outcome two regression coefficients are employed for covariate X1: one coefficient  $\beta_{11, \text{cat}_1} = \beta_{11, \text{cat}_2}$  for the first two categories and one coefficient  $\beta_{11, \text{cat}_3}$  for the third category. Covariate X3 only appears the model for the fourth outcome. For each of the two categories a different regression coefficient is estimated:  $\beta_{43, \text{cat}_1}$  and  $\beta_{43, \text{cat}_2}$ . Note also that some of the coefficients are constant across outcomes. Moreover the coefficient of covariate X5 is fixed to two by setting an offset in the model formula:

```
formula = 0 + X1 + X2 + X3 + X4 + offset(2 * X5) + X6 + X7
```

The constraints are set up as a named list where the names correspond to the names of all covariates in the `model.matrix`. The number of rows is equal to the total number of classes of the ordered responses, in the example above  $3 + 3 + 3 + 2 = 11$  rows. The number of columns represents the number of parameters to be estimated for each covariate.

```
coef.constraints = list(X1 = cbind(c(1, 1, rep(0, 9)), c(0, 0, 1, rep(0, 8)),
  c(rep(0, 3), rep(1, 3), rep(0, 5)), c(rep(0, 9), rep(1, 2))),
  X2 = cbind(c(rep(1, 9), rep(0, 2)), c(rep(0, 9), rep(0, 2))),
  X3 = cbind(c(rep(0, 9), 1, 0), c(rep(0, 9), 0, 1)),
  X4 = matrix(c(rep(1, 9), rep(0, 2))), X6 = cbind(c(rep(1, 3), rep(0, 8)),
  c(rep(0, 3), rep(1, 3), rep(0, 5)), c(rep(0, 6), rep(1, 3), rep(0, 2)),
  c(rep(0, 9), rep(1, 2))), X7 = cbind(c(rep(1, 3), rep(0, 8)),
  c(rep(0, 3), rep(1, 3), rep(0, 5)), c(rep(0, 6), rep(1, 3), rep(0, 2)),
  c(rep(0, 9), rep(1, 2))))
```

For more details we refer the reader to the documentation of the **VGAM** package.

### *Interaction terms and categorical covariates*

When constraints on the regression coefficients should be specified in models with interaction terms or categorical covariates, the `coef.constraints` matrix has to be constructed appropriately. If the order of the terms in the covariate matrix is not clear to the user, it is helpful to call the function `names_constraints()` before constructing the `coef.constraints` and `coef.values` matrices. Consider the following data:

	Y	X1	X2	X3	X4	X5
1	A	0.4341492	-0.8914832	c1	1.3081723	-0.1431266
2	B	0.6852053	-1.1332144	c2	0.1276609	1.1213270
3	A	1.0621450	1.5084641	c3	1.7144796	-1.0633038

where X3 is a categorical variable with three categories c1, c2 and c3. The names of each column in the covariate matrix can be obtained by:

```
R> formula <- Y ~ 1 + X1 : X2 + X3 + X4 + X3 * X5
R> names_constraints(formula, data)

[1] "(Intercept)" "X3c2"          "X3c3"          "X4"
[5] "X5"           "X1:X2"         "X3c2:X5"       "X3c3:X5"
```

or, without intercept:

```
R> formula <- Y ~ 0 + X1 : X2 + X3 + X4 + X3 * X5
R> names_constraints(formula, data)

[1] "X3c2"      "X3c3"      "X4"        "X5"        "X1:X2"     "X3c2:X5"
[7] "X3c3:X5"
```

This should be used when setting up the coefficient constraints. Please note that category `c1` for `X3` is taken as baseline for both cases.

### 3.7. Additional arguments

#### weights

Weights on each subject  $i$  can be chosen in a way that they are constant across multiple measurements. Weights should be stored in a column of `data`. The column name of the weights in `data` should be passed to this argument `weights`. Negative weights are not allowed.

#### offset

If offsets are not specified in the model `formula`, a list with a vector of offsets for each multiple measurement can be passed.

#### scale

If `scale = TRUE`, the continuous covariates are standardized by subtracting the mean and dividing by the standard deviation. This operation is performed for each repeated measurement before fitting.

#### solver

All general purpose optimizers of the R package **optimx** (Nash and Varadhan 2011; Nash 2014) can be used for maximization of the composite log-likelihood. These are "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlm", "nlminb", "spg", "ucminf", "newuoa", "bobyqa", "nmkb", "hjk", "Rcgmin" and "Rvmmin". The default is the "newuoa" solver. The "BFGS" solver performs well in terms of computation time, but it suffers from convergence problems, especially for the `mvlogit()` link. Moreover, if the user desires a specific solver which is not implemented in the R package **optimx**, other applicable solvers can be used by using a wrapper function with arguments `starting.values`, `objFun`, `control` of the following form:

```
solver = function(starting.values, objFun, control) {
  optRes <- solver.function(...)
  list(optimpar = optRes$optimpar, objvalue = optRes$objvalue)
}
```

The `solver.function()` should return a list of two elements `optimpar` and `objvalue`. The element `optimpar` should be a vector of length equal to number of parameters to optimize

containing the estimated parameters, while the element `objvalue` should contain the value of the objective function after the optimization procedure.

`se`

If `se = TRUE` standard errors are computed using the Godambe information matrix (see Section 2.4).

`start.values`

A list of starting values for threshold as well as regression coefficients can be passed by the argument `start.values`. This list contains a list (with a vector of starting values for each dimension) `theta` of all flexible threshold parameters and a list `beta` of all flexible regression parameters. All fixed values need to be excluded and in case of constraints on a whole dimension (e.g., `threshold.constraints = c(1,1,2,3)` or `coef.constraints = c(1,1,2,3)`), the element can be either skipped or a vector of length zero can be set. Starting values for Example 1 in Section 5 are for example:

```
start.values = list(theta = list(c(-3, -1, 0, 0.5, 2.5),
  c(-3, -1, 0, 0.5, 2, 3.5), c(0)), beta = list(c(0.05, -0.05, -0.8, 1, 0.2),
  c(-0.5, 0.2), c(-0.3, 0.3), c(0.5, -1.1, 0.7, 0.3, -1.2)))
```

Starting values for the error structure parameters cannot be specified by the user because of the complexity of the different error structure parameterizations and transformations.

`PL.lag`

In longitudinal studies, where  $q_i$  is possibly large, the pairwise likelihood estimation can be time consuming as it is build from all two dimensional combinations of  $j \in J_i$ . To overcome this difficulty, one can construct the likelihood using only the bivariate probabilities for pairs of observations less than  $lag$  in “time units” apart. A similar approach was proposed by [Varin and Czado \(2009\)](#). Assuming that, for each subject  $i$ , we have a time-series of consecutive ordinal observations, the  $i$ -th component of the pairwise likelihood has the following form:

$$p\ell_i^{lag}(\boldsymbol{\delta}) = w_i \left[ \sum_{k=1}^{q_i-1} \sum_{l=k+1}^{q_i} \mathbb{1}_{\{|t_k - t_l| \leq lag\}} \log P(Y_{ik} = r_{ik}, Y_{il} = r_{il}) \right].$$

The  $lag$  can be fixed by the argument `PL.lag` and it can only be used along with `error.structure = cor_ar1()`. The use of this argument is however not recommended if there are missing observations in the time series, i.e., if the ordinal variables are not observed in consecutive years. Moreover, one should also proceed with care if the observations are not missing at random.

`control`

A list of control arguments that are passed to the function `optimx()` or to the user-specific `solver.function()`. For further details see [Nash and Varadhan \(2011\)](#).

### 3.8. Methods for class ‘mvord’

Several methods are implemented for the class ‘mvord’. These methods include a `summary()` and a `print()` function to display the estimation results, a `coef()` function to extract the regression coefficients, a `thresholds()` function to extract the threshold coefficients and a function `get_error_struct()` to extract the estimated parameters of the correlation/covariance structure of the errors. The pairwise log-likelihood can be extracted by the function `logLik()`, function `vcov()` extracts the variance-covariance matrix of the parameters and `nobs()` provides the number of subjects. In addition, a `predict()` function, a `marginal.predict()` function and a `get.prob()` function are available to predict probabilities, cumulative probabilities, categories and linear predictors for the fitted model. Other standard methods such as `terms()`, `model.matrix()` and `fitted()` are also available. Functions `AIC()` and `BIC()` can be used to extract the information criteria CLAIC and CLBIC.

### 3.9. Output

The function `mvord()` returns an object of class ‘mvord’, which is a list containing the following components:

<code>beta</code>	a named <b>matrix</b> of regression coefficients
<code>theta</code>	a named <b>list</b> of threshold parameters
<code>error.struct</code>	an object of class ‘ <b>error_struct</b> ’
<code>sebeta</code>	a named <b>matrix</b> of standard errors of the regression coefficients
<code>setheta</code>	a named <b>list</b> of standard errors of the threshold parameters
<code>seerror.struct</code>	a <b>vector</b> of standard errors for the parameters of the error structure
<code>rho</code>	a <b>list</b> of all objects that are used in <code>mvord()</code>

### 3.10. Implementation `mvord2()`

Additionally, a second function `mvord2()` is implemented, for the setting where the covariates do not vary between the multiple measurements ( $\mathbf{x}_{i1} = \dots = \mathbf{x}_{iq}$ ):

```
mvord2(formula, data, error.structure = cor_general(~ 1), link = mvprobit()
  coef.constraints = NULL, coef.values = NULL, threshold.constraints = NULL,
  threshold.values = NULL, weights = NULL, se = TRUE, offset = NULL,
  scale = FALSE, start.values = NULL, solver = "newuoa", PL.lag = NULL,
  control = list(maxit = 200000, trace = 1, kkt = FALSE))
```

This function uses a slightly simplified data structure, where the multiple ordinal observations as well as the covariates are stored as columns in a **data.frame**. Each subject  $i$  corresponds to one row of the data frame, where all outcomes  $Y_{i1}, \dots, Y_{iq}$  (with missing observations set to NA) and all the covariates  $x_{i1}, \dots, x_{ip}$  are stored in different columns. Each outcome must be an ordered factor. In order to specify the model we use a multivariate **formula** object of the form:

```
formula = cbind(Y1, ..., Yq) ~ 0 + X1 + ... + Xp
```

The `error.structure` and the constraints on the regression and threshold parameters are set in analogy to `mvord()`. However, the ordering of the responses is given by the ordering in

the model `formula`. In addition, the `link`, subject `weights`, `se` and the `solver` are chosen in the same way as in `mvord()`.

## 4. Description of the illustrative data sets

As previously mentioned, the motivation of the **mvord** package lies in a credit risk application, where ordinal credit ratings from different rating sources are available for a sample of firms over several years. In this application, yearly credit ratings from the big three credit rating agencies Standard & Poor's, Moody's and Fitch were collected for a panel of publicly traded US companies. Moreover, accounting, stock price and default information on a yearly basis have been obtained for each firm-year in the sample.

Before proceeding with the multivariate analysis of these credit ratings, the following aspects of the problem at hand are worth mentioning. Firstly, a feature of the sample is the rather high number of missing values. Not all firms in the sample are rated by all three raters at the same point in time. In line with the market share the raters hold in the US ratings market, the number of rated firms per year differs between the three raters. Secondly, the raters can employ different rating scales in their rating methodology, e.g., different labeling of their ordinal rating classes or a different number of rating classes. A perhaps more fundamental difference between the rating scales appears when the raters employ different rating “philosophies”, i.e., when the central credit risk measure employed in assessing creditworthiness differs among raters (e.g., probability of default versus recovery in case of default). This also has implications for the covariates used in the rating models, as the inclusion or importance of various covariates can differ among the raters. Even though the rating agencies do not fully disclose their rating process, most likely for keeping their competitive advantage, they do publish reports on a regular basis where they state what are the main variables considered in the rating process.

These data have been analyzed in [Hirk \*et al.\* \(2017a\)](#), where a multivariate model of corporate credit ratings has been proposed. Unfortunately, this original data set is not freely redistributable. Therefore, we resort to the simulation of illustrative data sets by taking into consideration key features of the original data such as the ones mentioned above. The simulated data sets contain ratings from four different rating sources and firm-specific covariates for a panel of 1665 firms over a sample period of ten years.

### 4.1. Covariates

The covariates employed correspond to firm-level and market financial ratios which measure different aspects of a firm's financial health:

- ICR – interest coverage ratio, which measures how well the interest expenses can be covered from the free operating cash-flow of a company;
- LR – liquidity ratio relating the cash held by a company to the current liabilities;
- LEV1 – leverage ratio relating debt to earnings before interest and taxes;
- LEV2 – leverage ratio measuring the percentage of debt in the long-term capital of a firm;



- PR – profitability ratio measuring return on capital;
- 1RSIZE – relative size of the company in the market (in logarithm);
- 1SYSR – a measure of systematic risk (in logarithm);
- BSEC – the business sector of a firm.

More information on these covariates can be found in e.g., [Campbell, Hilscher, and Szilagyi \(2008\)](#) or [Puccia, Collett, Kernan, Palmer, Mettrick, and Deslondes \(2013\)](#). The business sector variable BSEC has been kept as in the original data set. For the continuous covariates we resort to simulation. As the distribution of the continuous covariates in the original data set is rather homogeneous across the ten years, we fit a distribution to each covariate over the whole sample period using the function `fitdistr()` of the **MASS** package. The best fitting distribution has been chosen by AIC. The parameters and distributions used for the simulation of the covariates are given in Table 2.

Covariate	Bound	Distribution	Parameters
ICR	unbounded	$t(m, s, df)$	$m = 2.301, s = 2.120, df = 1.163$
LR	positive	$\log-t(m, s, df)$	$m = -1.472, s = 0.992, df = 2.753$
LEV1	positive	$\log-t(m, s, df)$	$m = 1.008, s = 0.596, df = 1.245$
LEV2	positive	$\log-t(m, s, df)$	$m = -0.777, s = 0.443, df = 1.289$
PR	unbounded	$t(m, s, df)$	$m = 0.12, s = 0.08, df = 4$
1RSIZE	unbounded	$\mathcal{N}(m, s)$	$m = -9.131, s = 1.475$
1SYSR	unbounded	$\mathcal{N}(m, s)$	$m = -3.877, s = 0.439$

Table 2: Distributions used for the simulation of the covariates.

After simulating the covariates from the distributions presented in Table 2, we winsorize the simulated ratios at 10% for unbounded ratios and 5% for bounded ratios (by setting for all ratios the values above the 95%-quantile to the 95%-quantile; for the unbounded ratios we also winsorize from below, by setting for the values below the 5%-quantile to the 5%-quantile). This is standard practice in the literature on credit risk estimation based on financial ratios for reducing the influence of outliers (e.g., [Campbell et al. 2008](#)).

#### 4.2. Cross-sectional data set

For the 1665 firms in the first year of the sample, we generate ratings from four rating sources. We name the four rating sources R1, R2, R3 and R4. Raters R1, R2 and R3 correspond to the three rating agencies in the original application. The fourth rating source R4 corresponds to an indicator variable constructed from the original data set, which indicates if the company will be in an investment or speculative grade within one year from the rating observations. In the original data set, this indicator is computed by checking the one-year ahead rating for each of the three raters. We then follow the US regulation stating that, if two ratings for a company are available, one investment grade and one speculative grade, the company is categorized as speculative grade, whereas if three ratings are available the majority rule applies ([Bongaerts, Cremers, and Goetzmann 2012](#)).

Similarly to the original data, we assume that the rating methodology of the raters is heterogeneous. Raters R1 and R2 employ a similar credit risk measure when assigning firms on

a six-point ordinal scale (which we label, from best to worst, A, B, C, D, E, F), rater R3 uses a seven-point scale and different labeling (from best to worst, G, H, I, J, K, L, M) while the R4 ratings differentiate on a two point scale between speculative grade and investment grade firms (from best to worst, N, O).

We simulate the latent scores  $\tilde{Y}_{ij}$  from the model in Equation 1 with multivariate logit link and a constant general correlation structure. The correlation matrix used in the simulation is

$$\Sigma = \begin{pmatrix} 1 & 0.9 & 0.7 & 0.5 \\ 0.9 & 1 & 0.7 & 0.6 \\ 0.7 & 0.7 & 1 & 0.8 \\ 0.5 & 0.6 & 0.8 & 1 \end{pmatrix}.$$

We use the following coefficients:

- $\beta_{0j} = 0 \ \forall j \in \{\text{R1}, \text{R2}, \text{R3}, \text{R4}\};$
- $\beta_{\text{R1}} = (0.3, 0, -0.2, -1.3, 0.5, 0.2, 0)^\top;$
- $\beta_{\text{R2}} = (0, 0, -0.2, -1, 0.5, 0.3, 0)^\top;$
- $\beta_{\text{R3}} = (0.3, 0, -0.2, -1.4, 0.5, 0.4, 0)^\top;$
- $\beta_{\text{R4}} = (0, -0.3, 0, -1.6, 1.9, 0.1, -0.2)^\top.$

The threshold parameters are chosen such that the simulated distribution of ratings (the proportion of observations falling into each rating class) is similar to the distribution of the ratings in the original data set.

As raters R1 and R2 employ similar rating methodologies and use the same labeling of their rating classes, we set  $\theta_{\text{R1}} = \theta_{\text{R2}} = (-4.5, -3, -1, 1, 3.5)^\top$ . The thresholds for rater R3 are:  $\theta_{\text{R3}} = (-6, -3, 0, 1, 2, 4)^\top$  and the threshold for rater R4 is  $\theta_{\text{R4}} = -1$ .

The panel of ratings in the original data set is highly unbalanced, as not all firms receive ratings from all four sources. We therefore keep the missingness pattern and remove the simulated ratings that correspond to missing observations in the original data. For rater R1 we remove 5.41%, for rater R2 86.13%, and for rater R3 34.23% of the observations.

The cross-sectional data set is available in the **mvord** package as `data_cr_mvord` (in long format) or `data_cr_mvord2` (in short format). More details on the structure of this data set will be provided in Section 5.

### 4.3. Longitudinal data set

For the whole panel of firms we simulate ratings corresponding to rater R1 over a period of ten years (`year1`, `year2`, ..., `year10`). From the initial 1665 firms which are in the sample in the first year, 1575 are rated by rater R1.

We simulate the latent scores from the model in Equation 1 with probit link and business sector-specific  $AR(1)$  correlation structure. The dependence of the correlation structure on the business sector is motivated by the fact that in some sectors such as manufacturing ratings tend to be more “sticky”, i.e., do not change often over the years, while in more volatile sectors like IT there is less “stickiness” in the ratings. The sector specific correlations are:

$\rho_{\text{BSEC1}} = 0.9$ ,  $\rho_{\text{BSEC2}} = 0.7$ ,  $\rho_{\text{BSEC3}} = 0.9$ ,  $\rho_{\text{BSEC4}} = 0.9$ ,  $\rho_{\text{BSEC5}} = 0.9$ ,  $\rho_{\text{BSEC6}} = 0.7$ ,  $\rho_{\text{BSEC7}} = 0.5$ ,  $\rho_{\text{BSEC8}} = 0.6$ .

We use no intercepts and two sets of regression coefficients:

- $\beta_{\text{year1}} = \dots = \beta_{\text{year5}} = (0.17, 0, -0.11, -0.72, 0.28, 0.11, 0)^\top$ ;
- $\beta_{\text{year6}} = \dots = \beta_{\text{year10}} = (0.08, -0.2, -0.3, -1.6, 0.4, 0.5, -0.2)^\top$ .

Such a “break” in the coefficients could be explained by e.g., a change of regimes. Moreover, we assume the rating methodology of rater R1 does not change over time, hence we use one set of threshold parameters for all years:  $\theta_{\text{year1}} = \dots = \theta_{\text{year10}} = (-4, -1, 0, 1, 3)^\top$ . The longitudinal panel is unbalanced, as firms can leave the sample for various reasons such as default or mergers and acquisitions. In addition, credit ratings can be withdrawn. We remove the simulated ratings that correspond to missing firm-year observations in the original data. The longitudinal panel of R1 ratings together with the covariates is available in `data_cr_panel`.

## 5. Examples

We use the credit risk data sets presented in Section 4 to illustrate some of the features of package `mvord`.

### 5.1. Example 1 – Ratings assigned by multiple raters to a cross-section of firms

The first example presents a multivariate ordinal regression model with logit link and a general correlation error structure (`cor_general(~ 1)`). The simulated data set contains the credit risk measure `rating` (ratings assigned by raters R1, R2, R3 and R4) and 8 covariates for a cross-section of 1665 firms. The number of firm-ratings is 4566.

```
R> head(data_cr_mvord, n = 3)
```

	firm_id	rater_id	rating	ICR	LR	LEV1	LEV2
1	1	R1	D	1.546318	0.2484137	3.782934	0.92053787
2	2	R1	B	8.723779	0.1506502	1.033042	0.05305052
3	3	R1	D	4.726520	0.5187664	8.942818	0.97001785

  

	PR	LRSIZE	LSYSR	BSEC
1	0.2743184	-11.202807	-3.691023	BSEC3
2	0.1182763	-8.815116	-4.270618	BSEC3
3	0.2871493	-9.548691	-3.895642	BSEC6

```
R> str(data_cr_mvord, vec.len = 3)
```

```
'data.frame':      4566 obs. of  11 variables:
 $ firm_id : Factor w/ 1665 levels "1","2","3","4",...: 1 2 3 4 5 6 7 8 ...
 $ rater_id: Factor w/ 4 levels "R1","R2","R3",...: 1 1 1 1 1 1 1 1 ...
 $ rating  : chr  "D" "B" "D" ...
 $ ICR     : num  1.55 8.72 4.73 4.08 ...
```

```

$ LR      : num  0.248 0.151 0.519 0.168 ...
$ LEV1    : num  3.78 1.03 8.94 2.19 ...
$ LEV2    : num  0.9205 0.0531 0.97 2.8743 ...
$ PR      : num  0.2743 0.1183 0.2871 0.0821 ...
$ lRSIZE  : num  -11.2 -8.82 -9.55 -8.66 ...
$ lSYSR   : num  -3.69 -4.27 -3.9 -5.13 ...
$ BSEC    : Factor w/ 8 levels "BSEC1","BSEC2",...: 3 3 6 4 3 1 6 4 ...

```

The distribution of the ratings classes for the four raters is:

```

R> by(data_cr_mvord, data_cr_mvord$rater_id,
+      function(x) table(x$rating))

```

```
data_cr_mvord$rater_id: R1
```

```

  A  B  C  D  E  F
89 450 605 281 89 61

```

```
-----
data_cr_mvord$rater_id: R2
```

```

  A  B  C  D  E  F
12 74 79 51 9 6

```

```
-----
data_cr_mvord$rater_id: R3
```

```

  G  H  I  J  K  L  M
40 163 169 209 404 88 22

```

```
-----
data_cr_mvord$rater_id: R4
```

```

  N  O
1067 598

```

We include 7 financial ratios as covariates in a model without intercept by the formula:

```
formula = rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR
```

The subject index  $i$  is stored in the column `firm_id` and the multiple measurement index  $j$ , which indicates the rater, is given by `rater_id`:

```
index = c("firm_id", "rater_id")
```

An optional vector `response.names` is used to specify all the raters to be included in the model. The ordering of this vector is essential when constraints on the parameter set want to be imposed:

```
response.names = c("R1", "R2", "R3", "R4")
```

Due to the fact that the categories differ across raters we specify the `response.levels` by:

```
response.levels = list(rev(LETTERS[1:6]), rev(LETTERS[1:6]),
  rev(LETTERS[7:13]), rev(LETTERS[14:15]))
```

If no `response.levels` are passed, the natural ordering is used and could lead to an incorrect labeling. The rating classes assigned by the raters are here in order from worst to best indicating that lower values of the latent variables indicate lower creditworthiness or increased credit risk.

We fit a model to these data with the following features:

- We assume that R1 and R2 use the same rating scale by setting the following constraints on the threshold parameters:

```
threshold.constraints = c(1, 1, 2, 3)
```

- We assume that some covariates are equal for some raters. For example, we assume that the coefficient of ICR is equal for R1 and R3, or that the coefficients of LEV1 and PR are the same for the raters R1, R2 and R3. In addition, some of the regression coefficients are set to zero like ICR for R1 and R3, or LSYSR for the raters R1, R2 and R3. All the constraints above and some additional constraints are performed by the following restrictions on the regression coefficients by using the more flexible method:

```
coef.constraints = cbind(c(1, NA, 1, NA), c(NA, NA, NA, 1),
  c(1, 1, 1, NA), c(1, 2, 3, 4), c(1, 1, 1, 4), c(1, 2, 3, 4),
  c(NA, NA, NA, 1))
```

The NAs in `coef.constraints` are fixed to specific values. If no matrix `coef.values` is provided, the coefficients are set by default to zero automatically. This automatically generated `coef.values` matrix, looks like:

```
coef.values = cbind(c(NA, 0, NA, 0), c(0, 0, 0, NA), c(NA, NA, NA, 0),
  c(NA, NA, NA, NA), c(NA, NA, NA, NA), c(NA, NA, NA, NA),
  c(0, 0, 0, NA))
```

The specified `coef.constraints` together with `coef.values` give the following model:

$$\begin{aligned}\tilde{Y}_1 &= \beta_{11}\text{ICR} + \beta_{13}\text{LEV1} + \beta_{14}\text{LEV2} + \beta_{15}\text{PR} + \beta_{16}\text{LRSIZE}, \\ \tilde{Y}_2 &= \beta_{13}\text{LEV1} + \beta_{24}\text{LEV2} + \beta_{15}\text{PR} + \beta_{26}\text{LRSIZE}, \\ \tilde{Y}_3 &= \beta_{11}\text{ICR} + \beta_{13}\text{LEV1} + \beta_{34}\text{LEV2} + \beta_{15}\text{PR} + \beta_{36}\text{LRSIZE}, \\ \tilde{Y}_4 &= \beta_{42}\text{LR} + \beta_{44}\text{LEV2} + \beta_{45}\text{PR} + \beta_{46}\text{LRSIZE} + \beta_{47}\text{LSYSR}.\end{aligned}$$

As a link function we choose the multivariate logit link:

```
link = mvlogit()
```

For simplicity, we use a general correlation structure which is constant for all subjects:

```
error.structure = cor_general(~ 1)
```

In order to avoid numerical instabilities, we standardize our data for each rater:

```
R> covar_names <- c("ICR", "LR", "LEV1", "LEV2", "PR", "lRSIZE", "lSYSR")
R> data_cr_mvord_scaled <- do.call("rbind.data.frame",
+   by(data_cr_mvord, data_cr_mvord$rater_id,
+     function(x){x[, covar_names] <- scale(x[, covar_names]); x}))
```

The estimation can now be performed by the function `mvord()`:

```
R> res_cor_logit <- mvord(
+   formula = rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR,
+   data = data_cr_mvord_scaled, error.structure = cor_general(~ 1),
+   link = mvlogit(), index = c("firm_id", "rater_id"),
+   response.names = c("R1", "R2", "R3", "R4"),
+   response.levels = list(rev(LETTERS[1:6]), rev(LETTERS[1:6]),
+     rev(LETTERS[7:13]), rev(LETTERS[14:15])),
+   coef.constraints = cbind(c(1, NA, 1, NA), c(NA, NA, NA, 1),
+     c(1, 1, 1, NA), c(1, 2, 3, 4), c(1, 1, 1, 4), c(1, 2, 3, 4),
+     c(NA, NA, NA, 1)), threshold.constraints = c(1, 1, 2, 3),
+   solver = "newuoa")
```

The results are displayed by the function `summary()`:

```
R> summary(res_cor_logit, call = FALSE)
```

Formula: rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR

	link	threshold	nsubjects	ndim	logPL	CLAIC	CLBIC	fevals
mvlogit	flexible		1665	4	-7995.86	16073.9	16296.51	10115

Thresholds:

	Estimate	Std. Error	z value	Pr(> z )
R1 F E	-4.864072	0.116257	-41.8388	<2e-16 ***
R1 E D	-3.208460	0.094781	-33.8514	<2e-16 ***
R1 D C	-1.101554	0.048797	-22.5740	<2e-16 ***
R1 C B	0.946336	0.027355	34.5952	<2e-16 ***
R1 B A	3.394200	0.024222	140.1274	<2e-16 ***
R3 M L	-6.142840	0.611119	-10.0518	<2e-16 ***
R3 L K	-3.194905	0.038080	-83.9009	<2e-16 ***
R3 K J	0.026137	0.016829	1.5531	0.1204
R3 J I	1.060619	0.025063	42.3186	<2e-16 ***
R3 I H	2.048267	0.041325	49.5647	<2e-16 ***
R3 H G	3.979527	0.036695	108.4493	<2e-16 ***
R4 O N	-0.984248	0.027912	-35.2627	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
ICR 1	0.3545938	0.0094313	37.5977	< 2.2e-16 ***
LR 1	-0.3823069	0.0379644	-10.0701	< 2.2e-16 ***
LEV1 1	-0.1813132	0.0209321	-8.6620	< 2.2e-16 ***
LEV2 1	-1.3930198	0.0388631	-35.8443	< 2.2e-16 ***
LEV2 2	-1.0062239	0.0740580	-13.5870	< 2.2e-16 ***
LEV2 3	-1.4657429	0.0332794	-44.0435	< 2.2e-16 ***
LEV2 4	-1.7196085	0.0661440	-25.9979	< 2.2e-16 ***
PR 1	0.5533721	0.0247144	22.3906	< 2.2e-16 ***
PR 2	2.1773372	0.0227730	95.6103	< 2.2e-16 ***
LRSIZE 1	0.2535426	0.0139068	18.2315	< 2.2e-16 ***
LRSIZE 2	0.4103729	0.0719033	5.7073	1.148e-08 ***
LRSIZE 3	0.4754554	0.0278519	17.0708	< 2.2e-16 ***
LRSIZE 4	0.1655771	0.0147166	11.2511	< 2.2e-16 ***
LSYSR 1	-0.2013635	0.0121648	-16.5530	< 2.2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Error Structure:

	Estimate	Std. Error	z value	Pr(> z )
corr R1 R2	0.903644	0.021156	42.7131	< 2.2e-16 ***
corr R1 R3	0.693240	0.020956	33.0806	< 2.2e-16 ***
corr R1 R4	0.493521	0.053239	9.2699	< 2.2e-16 ***
corr R2 R3	0.734072	0.043160	17.0082	< 2.2e-16 ***
corr R2 R4	0.598471	0.112424	5.3234	1.019e-07 ***
corr R3 R4	0.770318	0.037239	20.6857	< 2.2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

If the user fixes thresholds to specific values, the  $z$  values and the corresponding  $p$  values as well as the significance codes are set to NA. Another option to display the results is the function `print()`:

```
R> print(res_cor_logit, call = FALSE)
```

## Thresholds:

\$R1

F E	E D	D C	C B	B A
-4.864072	-3.208460	-1.101554	0.946336	3.394200

\$R2

F E	E D	D C	C B	B A
-4.864072	-3.208460	-1.101554	0.946336	3.394200

\$R3



```

      M|L      L|K      K|J      J|I      I|H
-6.14283986 -3.19490513  0.02613705  1.06061876  2.04826658
      H|G
  3.97952741

```

```
$R4
```

```

      O|N
-0.984248

```

Coefficients:

```

      ICR 1      LR 1      LEV1 1      LEV2 1      LEV2 2      LEV2 3
  0.3545938 -0.3823069 -0.1813132 -1.3930198 -1.0062239 -1.4657429
      LEV2 4      PR 1      PR 2      lRSIZE 1      lRSIZE 2      lRSIZE 3
-1.7196085  0.5533721  2.1773372  0.2535426  0.4103729  0.4754554
      lRSIZE 4      lSYSR 1
  0.1655771 -0.2013635

```

Parameters of the error structure:

```

corr R1 R2 corr R1 R3 corr R1 R4 corr R2 R3 corr R2 R4 corr R3 R4
  0.9036439  0.6932396  0.4935211  0.7340721  0.5984705  0.7703181

```

An extended summary, where all thresholds and regression coefficients are shown, even though they are duplicated, can be obtained by:

```
R> summary(res_cor_logit, short = FALSE, call = FALSE)
```

The threshold coefficients can be extracted by the function `thresholds()`:

```
R> thresholds(res_cor_logit)
```

```
$R1
```

```

      F|E      E|D      D|C      C|B      B|A
-4.864072 -3.208460 -1.101554  0.946336  3.394200

```

```
$R2
```

```

      F|E      E|D      D|C      C|B      B|A
-4.864072 -3.208460 -1.101554  0.946336  3.394200

```

```
$R3
```

```

      M|L      L|K      K|J      J|I      I|H
-6.14283986 -3.19490513  0.02613705  1.06061876  2.04826658
      H|G
  3.97952741

```

```
$R4
```

```

      O|N
-0.984248

```

The regression coefficients are obtained by the function `coef()`:

```
R> coef(res_cor_logit)
```

```
      ICR 1      LR 1      LEV1 1      LEV2 1      LEV2 2      LEV2 3
0.3545938 -0.3823069 -0.1813132 -1.3930198 -1.0062239 -1.4657429
      LEV2 4      PR 1      PR 2      1RSIZE 1      1RSIZE 2      1RSIZE 3
-1.7196085  0.5533721  2.1773372  0.2535426  0.4103729  0.4754554
      1RSIZE 4      1SYSR 1
0.1655771 -0.2013635
```

The error structure for firm with `firm_id` 11 is displayed by the function `get_error_struct()`:

```
R> get_error_struct(res_cor_logit)[["11"]]
```

```
      R1      R2      R3      R4
R1 1.0000000 0.9036439 0.6932396 0.4935211
R2 0.9036439 1.0000000 0.7340721 0.5984705
R3 0.6932396 0.7340721 1.0000000 0.7703181
R4 0.4935211 0.5984705 0.7703181 1.0000000
```

*Fitting the model with the function `mvord2()`*

Due to the fact that the covariates do not change across the multiple measurements (the covariates are firm-specific and do not vary across raters), we can alternatively fit the model by the function `mvord2()`. In `mvord2()`, a slightly different format of `data` is used and the ordering of the responses is defined by a multivariate `formula` object. The multiple measurements are stored in different columns as ordered factors:

```
R> head(data_cr_mvord2, n = 3)
```

```
  firm_id R1  R2  R3 R4      ICR      LR      LEV1      LEV2
1      1  D <NA>   K  N 1.546318 0.2484137 3.782934 0.92053787
2      2  B <NA> <NA> N 8.723779 0.1506502 1.033042 0.05305052
3      3  D <NA> <NA> N 4.726520 0.5187664 8.942818 0.97001785
      PR      1RSIZE      1SYSR BSEC
1 0.2743184 -11.202807 -3.691023 BSEC3
2 0.1182763 -8.815116 -4.270618 BSEC3
3 0.2871493 -9.548691 -3.895642 BSEC6
```

```
R> str(data_cr_mvord2, vec.len = 2)
```

```
'data.frame':      1665 obs. of  13 variables:
 $ firm_id: Factor w/ 1665 levels "1","2","3","4",...: 1 2 3 4 5 ...
 $ R1      : Ord.factor w/ 6 levels "F"<"E"<"D"<"C"<...: 3 5 3 1 5 ...
```

```

$ R2      : Ord.factor w/ 6 levels "F"<"E"<"D"<"C"<...: NA NA NA NA NA ...
$ R3      : Ord.factor w/ 7 levels "M"<"L"<"K"<"J"<...: 3 NA NA 1 NA ...
$ R4      : Ord.factor w/ 2 levels "O"<"N": 2 2 2 1 2 ...
$ ICR     : num  1.55 8.72 ...
$ LR      : num  0.248 0.151 ...
$ LEV1    : num  3.78 1.03 ...
$ LEV2    : num  0.9205 0.0531 ...
$ PR      : num  0.274 0.118 ...
$ lRSIZE  : num  -11.2 -8.82 ...
$ lSYSR   : num  -3.69 -4.27 ...
$ BSEC    : Factor w/ 8 levels "BSEC1","BSEC2",...: 3 3 6 4 3 ...

```

Again, we standardize the data to avoid numerical instabilities:

```
R> data_cr_mvord2[, covar_names] <- scale(data_cr_mvord2[, covar_names])
```

The estimation is performed by calling the function `mvord2()`:

```

R> res_cor_logit <- mvord2(
+   formula = cbind(R1, R2, R3, R4) ~ 0 + ICR + LR + LEV1 + LEV2 + PR +
+     lRSIZE + lSYSR, error.structure = cor_general(~ 1), link = mvlogit(),
+   data = data_cr_mvord2, coef.constraints = cbind(c(1, NA, 1, NA),
+     c(NA, NA, NA, 1), c(1, 1, 1, NA), c(1, 2, 3, 4), c(1, 1, 1, 4),
+     c(1, 2, 3, 4), c(NA, NA, NA, 1)),
+   threshold.constraints = c(1, 1, 2, 3))

```

yielding equivalent results to the fit of `mvord()`.

## 5.2. Example 2 – Ratings assigned by one rater to a panel of firms

In a second example we present a longitudinal multivariate ordinal probit regression model with a covariate dependent  $AR(1)$  error structure. The simulated data set contains the credit risk measure `rating` (ratings assigned by rater R1) and 8 covariates for a panel of 1575 firms over ten years. The number of firm-year observations is 11037:

```
R> str(data_cr_panel, vec.len = 3)
```

```

'data.frame':      11037 obs. of  11 variables:
 $ firm_id: Factor w/ 1575 levels "1","2","3","4",...: 1 2 3 4 5 6 7 8 ...
 $ year   : Factor w/ 10 levels "year1","year2",...: 1 1 1 1 1 1 1 1 ...
 $ rating : Ord.factor w/ 6 levels "F"<"E"<"D"<"C"<...: 4 3 4 5 4 6 3 4 ...
 $ ICR    : num  2.9281 3.9104 3.1522 -0.0483 ...
 $ LR     : num  0.6433 0.3154 0.1471 0.0162 ...
 $ LEV1   : num  3.5 6.84 0.26 21.98 ...
 $ LEV2   : num  0.392 0.409 0.414 0.442 ...
 $ PR     : num  0.2075 0.05 0.0968 0.1764 ...
 $ lRSIZE : num  -10.69 -8.07 -8.64 -9.4 ...
 $ lSYSR  : num  -4.18 -3.22 -4.16 -4.21 ...
 $ BSEC   : Factor w/ 8 levels "BSEC1","BSEC2",...: 3 3 6 4 3 1 6 4 ...

```

```
R> head(data_cr_panel, n = 3)
```

	firm_id	year	rating	ICR	LR	LEV1	LEV2
1	1	year1	C	2.928058	0.6433189	3.5040486	0.3919582
2	2	year1	D	3.910441	0.3154287	6.8438839	0.4089648
3	3	year1	C	3.152185	0.1471216	0.2595343	0.4141279

  

	PR	1RSIZE	1SYSR	BSEC
1	0.20746812	-10.687212	-4.182475	BSEC3
2	0.05000904	-8.069681	-3.221042	BSEC3
3	0.09679941	-8.636322	-4.157882	BSEC6

The panel is highly unbalanced. The distribution of the number of ratings per firm assigned by rater R1 over the 10 years is given by:

```
R> summary(rowSums(with(data_cr_panel, table(firm_id, year))))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.000	4.000	9.000	7.008	10.000	10.000

Per year the number of ratings in the data set decreases:

```
R> with(data_cr_panel, table(year))
```

year	year1	year2	year3	year4	year5	year6	year7	year8	year9	year10
	1575	1424	1327	1208	1101	1020	924	867	811	780

We include the 7 financial ratios as covariates in a model without intercept by the **formula**:

```
formula = rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + 1RSIZE + 1SYSR
```

The subject index  $i$  is stored in the column `firm_id` while the multiple measurement index  $j$  is given in the column `year`:

```
index = c("firm_id", "year")
```

If we wish to estimate the model only for the last eight years of the sample, this can be done by specifying the names of each ordered response which should enter the model:

```
response.names = paste0("year", 3:10)
```

The rating classes assigned by rater R1 are:

```
R> levels(data_cr_panel$rating)
```

```
[1] "F" "E" "D" "C" "B" "A"
```

with the sixth rating class F being the worst class and the first rating class A being the best rating class. We specify the response levels, in the order from worst to best, for each of the eight outcome dimensions through the `response.level` argument. Ordering the classes from worst to best indicates that lower values of the latent variables indicate lower creditworthiness or increased credit risk. The rating classes and labels do not change over the eight years:

```
response.levels = rep(list(levels(data_cr_panel$rating)),
  length(response.names))
```

Additionally, the model has the following features:

- We assume that the rating agencies do not change their methodology over the sample period. This means the threshold parameters are constant over the years. This can be specified through the argument `threshold.constraints`:

```
threshold.constraints = rep(1, length(response.names))
```

- We assume that there is a break-point in the regression coefficients after `year5` in the sample. This break-point could correspond to the beginning of a crisis in a real case application. Hence, we use one set of regression coefficients for years `year3`, `year4` and `year5` and a different set for `year6`, `year7`, `year8`, `year9`, `year10`. This can be specified through the argument `coef.constraints`:

```
coef.constraints = c(rep(1, 3), rep(2, 5))
```

- We allow for different correlation parameters in the  $AR(1)$  structure for the different business sectors:

```
error.structure = cor_ar1(~ BSEC)
```

As before, we standardize our covariates on a yearly basis:

```
R> data_cr_panel_scaled <- do.call("rbind.data.frame",
+   by(data_cr_panel, data_cr_panel$year,
+     function(x){x[, covar_names] <- scale(x[, covar_names]); x}))
```

The estimation is performed by calling the function `mvord()`:

```
R> res_AR1_probit <- mvord(
+   formula = rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR,
+   error.structure = cor_ar1(~ BSEC), link = mvprobit(),
+   index = c("firm_id", "year"), data = data_cr_panel_scaled,
+   response.levels = rep(list(levels(data_cr_panel$rating)), 8),
+   response.names = paste0("year", 3:10), coef.constraints = c(rep(1, 3),
+     rep(2, 5)), threshold.constraints = rep(1, 8), solver = "BFGS")
```

The results are displayed either by the function `summary()`:

```
R> summary(res_AR1_probit, short = TRUE, call = FALSE, digits = 6)
```

Formula: rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR

	link	threshold	nsubjects	ndim	logPL	CLAIC	CLBIC	fevals
mvprobit	flexible		1341	8	-56318.25	112935.33	113712.46	5672

Thresholds:

		Estimate	Std. Error	z value	Pr(> z )
year3	F E	-3.355647	0.071370	-47.0178	<2e-16 ***
year3	E D	-0.988439	0.031103	-31.7792	<2e-16 ***
year3	D C	-0.022681	0.026074	-0.8699	0.3844
year3	C B	0.959397	0.029453	32.5743	<2e-16 ***
year3	B A	2.799414	0.059097	47.3701	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Coefficients:

		Estimate	Std. Error	z value	Pr(> z )
ICR	1	0.096207	0.010211	9.4217	< 2.2e-16 ***
ICR	2	0.099580	0.028776	3.4605	0.0005391 ***
LR	1	-0.099299	0.010007	-9.9231	< 2.2e-16 ***
LR	2	-0.198422	0.030024	-6.6087	3.877e-11 ***
LEV1	1	-0.201947	0.010632	-18.9940	< 2.2e-16 ***
LEV1	2	-0.264218	0.031248	-8.4555	< 2.2e-16 ***
LEV2	1	-1.010343	0.018436	-54.8040	< 2.2e-16 ***
LEV2	2	-1.398467	0.043710	-31.9939	< 2.2e-16 ***
PR	1	0.317577	0.011144	28.4972	< 2.2e-16 ***
PR	2	0.346879	0.029859	11.6171	< 2.2e-16 ***
lRSIZE	1	0.304745	0.011136	27.3667	< 2.2e-16 ***
lRSIZE	2	0.495564	0.030856	16.0603	< 2.2e-16 ***
lSYSR	1	-0.101745	0.010265	-9.9119	< 2.2e-16 ***
lSYSR	2	-0.180581	0.029821	-6.0555	1.400e-09 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Error Structure:

		Estimate	Std. Error	z value	Pr(> z )
(Intercept)		1.066484	0.049278	21.6421	< 2.2e-16 ***
BSECBSEC2		-0.448659	0.067111	-6.6853	2.304e-11 ***
BSECBSEC3		0.133171	0.061238	2.1747	0.029656 *
BSECBSEC4		0.161037	0.056509	2.8498	0.004375 **
BSECBSEC5		0.140920	0.068992	2.0426	0.041096 *
BSECBSEC6		-0.327189	0.073677	-4.4408	8.961e-06 ***
BSECBSEC7		-0.590010	0.064416	-9.1593	< 2.2e-16 ***
BSECBSEC8		-0.574658	0.080068	-7.1771	7.121e-13 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

or by the function `print()`:

```
R> print(res_AR1_probit, call = FALSE, digits = 4)
```

Thresholds:

\$year3

F E	E D	D C	C B	B A
-3.35565	-0.98844	-0.02268	0.95940	2.79941

\$year4

F E	E D	D C	C B	B A
-3.35565	-0.98844	-0.02268	0.95940	2.79941

\$year5

F E	E D	D C	C B	B A
-3.35565	-0.98844	-0.02268	0.95940	2.79941

\$year6

F E	E D	D C	C B	B A
-3.35565	-0.98844	-0.02268	0.95940	2.79941

\$year7

F E	E D	D C	C B	B A
-3.35565	-0.98844	-0.02268	0.95940	2.79941

\$year8

F E	E D	D C	C B	B A
-3.35565	-0.98844	-0.02268	0.95940	2.79941

\$year9

F E	E D	D C	C B	B A
-3.35565	-0.98844	-0.02268	0.95940	2.79941

\$year10

F E	E D	D C	C B	B A
-3.35565	-0.98844	-0.02268	0.95940	2.79941

Coefficients:

ICR 1	ICR 2	LR 1	LR 2	LEV1 1	LEV1 2	LEV2 1
0.09621	0.09958	-0.09930	-0.19842	-0.20195	-0.26422	-1.01034
LEV2 2	PR 1	PR 2	1RSIZE 1	1RSIZE 2	1SYSR 1	1SYSR 2
-1.39847	0.31758	0.34688	0.30475	0.49556	-0.10175	-0.18058

Parameters of the error structure:

(Intercept)	BSECBSEC2	BSECBSEC3	BSECBSEC4	BSECBSEC5
1.0665	-0.4487	0.1332	0.1610	0.1409
BSECBSEC6	BSECBSEC7	BSECBSEC8		



-0.3272      -0.5900      -0.5747

An extended summary, where all thresholds and regression coefficients are shown, even though they are duplicated, can be obtained by:

```
R> summary(res_AR1_probit, short = FALSE, call = FALSE)
```

The threshold coefficients can be extracted by the function `thresholds()`:

```
R> thresholds(res_AR1_probit)
```

\$year3

	F E	E D	D C	C B	B A
	-3.35564738	-0.98843881	-0.02268105	0.95939737	2.79941429

\$year4

	F E	E D	D C	C B	B A
	-3.35564738	-0.98843881	-0.02268105	0.95939737	2.79941429

\$year5

	F E	E D	D C	C B	B A
	-3.35564738	-0.98843881	-0.02268105	0.95939737	2.79941429

\$year6

	F E	E D	D C	C B	B A
	-3.35564738	-0.98843881	-0.02268105	0.95939737	2.79941429

\$year7

	F E	E D	D C	C B	B A
	-3.35564738	-0.98843881	-0.02268105	0.95939737	2.79941429

\$year8

	F E	E D	D C	C B	B A
	-3.35564738	-0.98843881	-0.02268105	0.95939737	2.79941429

\$year9

	F E	E D	D C	C B	B A
	-3.35564738	-0.98843881	-0.02268105	0.95939737	2.79941429

\$year10

	F E	E D	D C	C B	B A
	-3.35564738	-0.98843881	-0.02268105	0.95939737	2.79941429

The regression coefficients are obtained by the function `coef()`:

```
R> coef(res_AR1_probit)
```

```

      ICR 1      ICR 2      LR 1      LR 2      LEV1 1
0.09620694 0.09957962 -0.09929923 -0.19842194 -0.20194715
      LEV1 2      LEV2 1      LEV2 2      PR 1      PR 2
-0.26421791 -1.01034302 -1.39846675 0.31757691 0.34687875
      lRSIZE 1      lRSIZE 2      lSYSR 1      lSYSR 2
0.30474549 0.49556382 -0.10174521 -0.18058107

```

The error structure is displayed by the function `get_error_struct()`:

```
R> get_error_struct(res_AR1_probit)
```

```

(Intercept)  BSECBSEC2  BSECBSEC3  BSECBSEC4  BSECBSEC5
1.0664840   -0.4486589   0.1331714   0.1610373   0.1409199
BSECBSEC6    BSECBSEC7  BSECBSEC8
-0.3271891   -0.5900100  -0.5746576

```

In addition, the correlation parameters  $\rho_i$  for each firm are obtained by:

```
R> head(get_error_struct(res_AR1_probit, type = "corr"), n = 3)
```

```

Correlation
1  0.8335495
2  0.8335495
3  0.6287190

```

Moreover, the correlation matrices for each specific firm are obtained by:

```
R> head(get_error_struct(res_AR1_probit, type = "sigmas"), n = 1)
```

```

$`1`
      year3      year4      year5      year6      year7      year8
year3 1.0000000 0.8335495 0.6948047 0.5791541 0.4827536 0.4023990
year4 0.8335495 1.0000000 0.8335495 0.6948047 0.5791541 0.4827536
year5 0.6948047 0.8335495 1.0000000 0.8335495 0.6948047 0.5791541
year6 0.5791541 0.6948047 0.8335495 1.0000000 0.8335495 0.6948047
year7 0.4827536 0.5791541 0.6948047 0.8335495 1.0000000 0.8335495
year8 0.4023990 0.4827536 0.5791541 0.6948047 0.8335495 1.0000000
year9 0.3354195 0.4023990 0.4827536 0.5791541 0.6948047 0.8335495
year10 0.2795887 0.3354195 0.4023990 0.4827536 0.5791541 0.6948047
      year9      year10
year3 0.3354195 0.2795887
year4 0.4023990 0.3354195
year5 0.4827536 0.4023990
year6 0.5791541 0.4827536
year7 0.6948047 0.5791541
year8 0.8335495 0.6948047
year9 1.0000000 0.8335495
year10 0.8335495 1.0000000

```

## 6. Conclusion

The present paper is meant to provide a general overview on the R package **mvord**, which implements the estimation of multivariate ordered probit and logit regression models using the pairwise likelihood approach. Different error structures like a general correlation and covariance structures, a covariate dependent equicorrelation structure and an  $AR(1)$  structure can be imposed. In addition, the flexible modeling framework allows imposing constraints on threshold as well as regression coefficients. Two different data formats can be used either by applying `mvord()` or `mvord2()` to estimate the model parameters.

Further research and possible extensions of **mvord** could be addressed to the implementation of variable selection procedures in multivariate ordinal regression models and the inclusion of multivariate semi- or nonparametric ordinal models.

## References

- Afonso A, Gomes P, Rother P (2009). “Ordered Response Models for Sovereign Debt Ratings.” *Applied Economics Letters*, **16**(8), 769–773. doi:10.1080/13504850701221931.
- Agresti A (2002). *Categorical Data Analysis*. 2nd edition. John Wiley & Sons.
- Agresti A (2010). *Analysis of Ordinal Categorical Data*. 2nd edition. John Wiley & Sons.
- Alp A (2013). “Structural Shifts in Credit Rating Standards.” *The Journal of Finance*, **68**(6), 2435–2470. doi:10.1111/jofi.12070.
- Bhat CR, Varin C, Ferdous N (2010). “A Comparison of the Maximum Simulated Likelihood and Composite Marginal Likelihood Estimation Approaches in the Context of the Multivariate Ordered-Response Model.” In *Maximum Simulated Likelihood Methods and Applications*, pp. 65–106. Emerald Group Publishing Limited. doi:10.1108/S0731-9053(2010)0000026007.
- Blume ME, Lim F, Mackinlay AC (1998). “The Declining Credit Quality of U.S. Corporate Debt: Myth or Reality?” *The Journal of Finance*, **53**(4), 1389–1413. doi:10.1111/0022-1082.00057.
- Boes S, Winkelmann R (2006). “Ordered Response Models.” *AStA Advances in Statistical Analysis*, **90**(1), 167–181. URL <http://EconPapers.repec.org/RePEc:spr:alstar:v:90:y:2006:i:1:p:167-181>.
- Bongaerts D, Cremers KJM, Goetzmann WN (2012). “Tiebreaker: Certification and Multiple Credit Ratings.” *The Journal of Finance*, **67**(1), 113–152. doi:10.1111/j.1540-6261.2011.01709.x.
- 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.
- Campbell JY, Hilscher J, Szilagyi J (2008). “In Search of Distress Risk.” *The Journal of Finance*, **63**(6), 2899–2939. doi:10.3386/w12362.

- Carroll N (2016). **oglmx**: *Estimation of Ordered Generalized Linear Models*. R package version 2.0.0.1, URL <https://CRAN.R-project.org/package=oglmx>.
- Christensen RHB (2015a). *Analysis of Ordinal Data with Cumulative Link Models - Estimation with the R-Package ordinal*. URL <https://CRAN.R-project.org/package=ordinal>.
- Christensen RHB (2015b). “**ordinal** – Regression Models for Ordinal Data.” R package version 2015.6-28, URL <https://CRAN.R-project.org/package=ordinal>.
- Fahrmeir L, Tutz G (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models*. 2nd edition. Springer-Verlag.
- Genz A, Bretz F (2009). *Computation of Multivariate Normal and t Probabilities*. Springer-Verlag.
- Genz A, Kenkel B (2015). **pbivnorm**: *Vectorized Bivariate Normal CDF*. R package version 0.6.0, URL <https://CRAN.R-project.org/package=pbivnorm>.
- Greene WH, Hensher DA (2010). *Modeling Ordered Choices: A Primer*. Cambridge University Press.
- Gumbel EJ (1961). “Bivariate Logistic Distributions.” *Journal of the American Statistical Association*, **56**(294), 335–349. doi:10.1080/01621459.1961.10482117.
- Harrell Jr FE (2017). **rms**: *Regression Modeling Strategies*. R package version 5.1-1, URL <https://CRAN.R-project.org/package=rms>.
- Hedeker D, Archer KJ, Nordgren R, Gibbons RD (2015). **mixor**: *Mixed-Effects Ordinal Regression Analysis*. R package version 1.0.3, URL <https://CRAN.R-project.org/package=mixor>.
- Higham NJ (1988). “Computing a Nearest Symmetric Positive Semidefinite Matrix.” *Linear Algebra and its Applications*, **103**, 103–118. doi:10.1016/0024-3795(88)90223-6.
- Hirk R, Hornik K, Vana L (2017a). “Multivariate Ordinal Regression Models: An Analysis of Corporate Credit Ratings.” *Research Report Series / Department of Statistics and Mathematics 132*, WU Vienna University of Economics and Business, Vienna. URL <http://epub.wu.ac.at/5389/>.
- Hirk R, Hornik K, Vana L (2017b). **mvord**: *An R Package for Fitting Multivariate Ordinal Regression Models*. R package version 0.1.0, URL <https://CRAN.R-project.org/package=mvord>.
- Hoetker G (2007). “The Use of Logit and Probit Models in Strategic Management Research: Critical Issues.” *Strategic Management Journal*, **28**(4), 331–343. doi:10.1002/smj.582.
- Kenne Pagui EC, Canale A (2016). “Pairwise Likelihood Inference for Multivariate Ordinal Responses with Applications to Customer Satisfaction.” *Applied Stochastic Models in Business and Industry*, **32**(2), 273–282. doi:10.1002/asmb.2147.
- Kenne Pagui EC, Canale A, Genz A, Azzalini A (2014). **PLordprob**: *Multivariate Ordered Probit Model via Pairwise Likelihood*. R package version 1.0, URL <https://CRAN.R-project.org/package=PLordprob>.

- Liu X (2009). “Ordinal Regression Analysis: Fitting the Proportional Odds Model Using Stata, SAS and SPSS.” *Journal of Modern Applied Statistical Methods*, **8**(2), 30. doi:[10.22237/jmasm/1257035340](https://doi.org/10.22237/jmasm/1257035340).
- Malik HJ, Abraham B (1973). “Multivariate Logistic Distributions.” *The Annals of Statistics*, **1**(3), 588–590. doi:[10.1214/aos/1176342430](https://doi.org/10.1214/aos/1176342430).
- 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](https://doi.org/10.18637/jss.v042.i09).
- McCullagh P (1980). “Regression Models for Ordinal Data.” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 109–142. URL <http://www.jstor.org/stable/2984952>.
- Nash JC (2014). “On Best Practice Optimization Methods in R.” *Journal of Statistical Software*, **60**(2), 1–14. doi:[10.18637/jss.v060.i02](https://doi.org/10.18637/jss.v060.i02).
- Nash JC, Varadhan R (2011). “Unifying Optimization Algorithms to Aid Software System Users: **optimx** for R.” *Journal of Statistical Software*, **43**(9), 1–14. doi:[10.18637/jss.v043.i09](https://doi.org/10.18637/jss.v043.i09).
- Nooraee N, Abegaz F, Ormel J, Wit E, van den Heuvel ER (2016). “An Approximate Marginal Logistic Distribution for the Analysis of Longitudinal Ordinal Data.” *Biometrics*, **72**(1), 253–261. doi:[10.1111/biom.12414](https://doi.org/10.1111/biom.12414).
- O’Brien SM, Dunson DB (2004). “Bayesian Multivariate Logistic Regression.” *Biometrics*, **60**(3), 739–746. doi:[10.1111/j.0006-341X.2004.00224.x](https://doi.org/10.1111/j.0006-341X.2004.00224.x).
- 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://tel.archives-ouvertes.fr/tel-01100921>.
- Peterson B, Harrell FE (1990). “Partial Proportional Odds Models for Ordinal Response Variables.” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **39**(2), 205–217. doi:[10.2307/2347760](https://doi.org/10.2307/2347760).
- Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2017). **nlme**: *Linear and Nonlinear Mixed Effects Models*. R package version 3.1-131, URL <https://CRAN.R-project.org/package=nlme>.
- Pinheiro JC, Bates DM (1996). “Unconstrained Parametrizations for Variance-Covariance Matrices.” *Statistics and Computing*, **6**(3), 289–296. doi:[10.1007/BF00140873](https://doi.org/10.1007/BF00140873).
- Python Software Foundation (2017). *Python Language Reference, Version 2.7*. Wilmington, DE. URL <http://www.python.org/>.
- R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- SAS Institute Inc (2017). *SAS/STAT Software, Version 9.4*. SAS Institute Inc., Cary, NC. URL <http://www.sas.com/>.

- SPSS Inc (2017). *IBM SPSS Statistics 24*. SPSS Inc., Chicago, IL. URL <http://www.spss.com/>.
- Puccia M, Collett LA, Kernan P, Palmer AD, Mettrick MS, Deslondes G (2013). “Request for Comment: Corporate Criteria.” *Technical report*, Standard and Poor’s Rating Services.
- Reusens P, Croux C (2017). “Sovereign Credit Rating Determinants: A Comparison before and after the European Debt Crisis.” *Journal of Banking & Finance*, **77**, 108–121. doi:[10.1016/j.jbankfin.2017.01.006](https://doi.org/10.1016/j.jbankfin.2017.01.006).
- Scott DM, Kanaroglou PS (2002). “An Activity-Episode Generation Model that Captures Interactions between Household Heads: Development and Empirical Analysis.” *Transportation Research Part B: Methodological*, **36**(10), 875 – 896. doi:[10.1016/S0191-2615\(01\)00039-X](https://doi.org/10.1016/S0191-2615(01)00039-X).
- StataCorp (2017). *Stata Statistical Software: Release 15*. StataCorp LLC, College Station, TX. URL <https://www.stata.com/>.
- Tutz G (2012). *Regression for Categorical Data*. Cambridge University Press.
- Varin C (2008). “On Composite Marginal Likelihoods.” *AStA Advances in Statistical Analysis*, **92**(1), 1. ISSN 1863-818X. doi:[10.1007/s10182-008-0060-7](https://doi.org/10.1007/s10182-008-0060-7).
- Varin C, Czado C (2009). “A Mixed Autoregressive Probit Model for Ordinal Longitudinal Data.” *Biostatistics*, pp. 1–12. doi:[10.1093/biostatistics/kxp042](https://doi.org/10.1093/biostatistics/kxp042).
- Varin C, Reid N, Firth D (2011). “An Overview of Composite Likelihood Methods.” *Statistica Sinica*, **21**(1), 5–42. URL <http://www.jstor.org/stable/24309261>.
- Varin C, Vidoni P (2005). “A Note on Composite Likelihood Inference and Model Selection.” *Biometrika*, **92**(3), 519–528. doi:[10.1093/biomet/92.3.519](https://doi.org/10.1093/biomet/92.3.519).
- Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York. URL <http://www.stats.ox.ac.uk/pub/MASS4>.
- Wurm M, Rathouz P, Hanlon B (2017). **ordinalNet**: *Penalized Ordinal Regression*. R package version 2.1, URL <https://CRAN.R-project.org/package=ordinalNet>.
- Yee TW (2010). “The **VGAM** Package for Categorical Data Analysis.” *Journal of Statistical Software*, **32**(10), 1–34. doi:[10.18637/jss.v032.i10](https://doi.org/10.18637/jss.v032.i10).

**Affiliation:**

Rainer Hirk  
Department of Finance, Accounting and Statistics  
Institute for Statistics and Mathematics  
WU Wirtschaftsuniversität Wien  
1020 Vienna, Austria  
E-mail: [rainer.hirk@wu.ac.at](mailto:rainer.hirk@wu.ac.at)

Kurt Hornik  
Department of Finance, Accounting and Statistics  
Institute for Statistics and Mathematics  
WU Wirtschaftsuniversität Wien  
1020 Vienna, Austria  
E-mail: [kurt.hornik@wu.ac.at](mailto:kurt.hornik@wu.ac.at)

Laura Vana  
Department of Finance, Accounting and Statistics  
Institute for Statistics and Mathematics  
WU Wirtschaftsuniversität Wien  
1020 Vienna, Austria  
E-mail: [laura.vana@wu.ac.at](mailto:laura.vana@wu.ac.at)