Package ‘aldvmm’

July 19, 2021

Type Package
Title Adjusted Limited Dependent Variable Mixture Models
Version 0.8.4
Date 2021-07-18
Depends R (>= 3.5.0)
Description The goal of the package ‘aldvmm’ is to fit adjusted limited dependent variable mixture models of health state utilities. Adjusted limited dependent variable mixture models are finite mixtures of normal distributions with an accumulation of density mass at the limits, and a gap between 100% quality of life and the next smaller utility value. The package ‘aldvmm’ uses the likelihood and expected value functions proposed by Hernandez Alava and Wailoo (2015) <doi:10.1177/1536867X1501500307> using normal component distributions and a multinomial logit model of probabilities of component membership.
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Imports numDeriv, stats, checkmate, optimr
URL https://github.com/pletschm/aldvmm/
BugReports https://github.com/pletschm/aldvmm/issues
Suggests knitr, markdown, tinytex, testthat (>= 3.0.0), covr, rmarkdown, bookdown, xtable, ggplot2, scales, reshape2
VignetteBuilder knitr
Config/testthat/edition 3
NeedsCompilation no
Author Mark Pletscher [aut, cre, cph]
Maintainer Mark Pletscher <pletscher.mark@gmail.com>
Repository CRAN
Date/Publication 2021-07-19 18:00:02 UTC
R topics documented:

aldvmm-package .................................................. 2
aldvmm ............................................................. 3
aldvmm.check ..................................................... 7
aldvmm.cv ........................................................... 9
aldvmm.getnames .................................................. 11
aldvmm.getpar ..................................................... 12
aldvmm.gof ........................................................... 13
aldvmm.gr .............................................................. 14
aldvmm.init ........................................................... 15
aldvmm.ll .............................................................. 17
aldvmm.mm ............................................................ 19
aldvmm.pred ........................................................... 20
aldvmm.sefit .......................................................... 21
aldvmm.sum ........................................................... 23
predict.aldvmm ...................................................... 24
summary.aldvmm ..................................................... 25
utility ................................................................. 26

Index 27

aldvmm-package  
aldvmm: Adjusted Limited Dependent Variable Mixture Models

Description

The goal of the package ‘aldvmm’ is to fit adjusted limited dependent variable mixture models of health state utilities. Adjusted limited dependent variable mixture models are finite mixtures of normal distributions with an accumulation of density mass at the limits, and a gap between 100% quality of life and the next smaller utility value. The package ‘aldvmm’ uses the likelihood and expected value functions proposed by Hernandez Alava and Wailoo (2015) <doi:10.1177/1536867X1501500307> using normal component distributions and a multinomial logit model of probabilities of component membership.

Examples

data(utility)

fit <- aldvmm(eq5d ~ age + female | 1,
data = utility,
psi = c(0.883, -0.594),
ncmp = 2)

summary(fit)

yhat <- predict(fit,
newdata = utility)
Description

The function `aldvmm()` fits adjusted limited dependent variable mixture models of health state utilities. Adjusted limited dependent variable mixture models are finite mixtures of normal distributions with an accumulation of density mass at the limits, and a gap between 100% quality of life and the next smaller utility value. The package `aldvmm` uses the likelihood and expected value functions proposed by Hernandez-Alava and Wailoo (2015) using normal component distributions and a multinomial logit model of probabilities of component membership.

Usage

```r
aldvmm(
  formula, 
  data, 
  psi, 
  ncmp = 2, 
  dist = "normal", 
  optim.method = NULL, 
  optim.control = list(trace = FALSE), 
  optim.grad = TRUE, 
  init.method = "zero", 
  init.est = NULL, 
  init.lo = NULL, 
  init.hu = NULL, 
  se.fit = FALSE, 
  level = 0.95
)
```

Arguments

- `formula` an object of class "formula" with a symbolic description of the model to be fitted. The model formula takes the form `y ~ x1 + x2 | x1 + x4`, where the | delimiter separates the model for expected values of normal distributions (left) and the multinomial logit model of probabilities of component membership (right).
- `data` a data frame, list or environment (or object coercible to a data frame by `base::as.data.frame()`) including data on outcomes and explanatory variables in 'formula'.
- `psi` a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. `c(-0.594, 0.883)`). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in 'psi' does not matter.
ncmp  a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.

dist  an optional character value of the distribution used in the finite mixture. In this release, only the normal distribution is available, and the default value is set to "normal".

optim.method  an optional character value of one of the following optimr::optimr() methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "Nelder-Mead". The method "L-BFGS-B" is used when lower and/or upper constraints are set using 'init.lo' and 'init.hi'. The method "nlm" cannot be used in the 'aldvmm' package.

optim.control  an optional list of optimr::optimr() control parameters.

optim.grad  an optional logical value indicating if a numerical gradient should be used in optimr::optimr() methods that can use this information. The default value is TRUE. If 'optim.grad' is set to FALSE, a finite difference approximation is used.

init.method  an optional character value indicating the method for obtaining initial values. The following values are available: "zero", "random", "constant" and "sann". The default value is "zero".

init.est  an optional numeric vector of user-defined initial values. User-defined initial values override the 'init.method' argument. Initial values have to follow the same order as parameter estimates in the return value 'par'.

init.lo  an optional numeric vector of user-defined lower limits for constrained optimization. When 'init.lo' is not NULL, the optimization method "L-BFGS-B" is used. Lower limits of parameters have to follow the same order as parameter estimates in the return value 'par'.

init.hi  an optional numeric vector of user-defined upper limits for constrained optimization. When 'init.hi' is not NULL, the optimization method "L-BFGS-B" is used. Upper limits of parameters have to follow the same order as parameter estimates in the return value 'par'.

se.fit  an optional logical value indicating whether standard errors of fitted values are calculated. The default value is FALSE.

level  a numeric value of the significance level for confidence bands of fitted values. The default value is 0.95.

Details

aldvmm() fits an adjusted limited dependent variable mixture model using the likelihood and expected value functions from Hernandez Alava and Wailoo (2015). The model accounts for latent classes, multi-modality, minimum and maximum utility values and potential gaps between 1 and the next smaller utility value. Adjusted limited dependent variable mixture models combine multiple component distributions with a multinomial logit model of the probabilities of component membership. The standard deviations of normal distributions are estimated and reported as log-transformed values which enter the likelihood function as exponentiated values to ensure non-negative values. The minimum utility and the largest utility smaller than or equal to 1 are supplied in the argument 'psi'. The number of distributions/components that are mixed is set by the argument 'ncmp'.

aldvmm
When `ncmp` is set to 1 the procedure estimates a tobit model with a gap between 1 and the maximum utility value in \( \psi \). The current version only allows finite mixtures of normal distributions.

The `formula` object can include a | delimiter to separate formulae for expected values in components (left) and the multinomial logit model of probabilities of group membership (right). If no | delimiter is used, the same formula will be used for expected values in components and the multinomial logit of the probabilities of component membership.

`aldvmm()` uses `optimr::optimr()` for maximum likelihood estimation of model parameters. The argument `optim.method` accepts the following methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "Nelder-Mead". The method "nlm" cannot be used in `aldvmm()` because it requires a different implementation of the likelihood function. The argument `optim.control` accepts a list of `optimr::optimr()` control parameters. If `optim.grad` is set to TRUE the function `optimr::optimr()` uses numerical gradients during the optimization procedure for all methods that allow for this approach. If `optim.grad` is set to FALSE or a method cannot use gradients, a finite difference approximation is used. The numerical gradients of the likelihood function are approximated numerically using the function `numDeriv::grad()`. The hessian matrix at maximum likelihood parameters is approximated numerically using `numDeriv::hessian()`.

`init.method` accepts four values of methods for generating initial values: "zero", "random", "constant", "sann". The method "zero" sets initial values of all parameters to 0. The method "random" draws random starting values from a standard normal distribution. The method "constant" estimates a constant-only model and uses estimates as initial values of intercepts and standard errors and 0 for all other parameters. The method "sann" estimates the full model using the simulated annealing optimization method in `stats::optim()` and uses parameter estimates as initial values. When user-specified initial values are supplied in `init.est`, the argument `init.method` is ignored.

By default, `aldvmm()` performs unconstrained optimization with upper and lower limits at -Inf and Inf. When user-defined lower and upper limits are supplied to `init.lo` and/or `init.hi`, these default limits are replaced with the user-specified values, and the method "L-BFGS-B" is used for box-constrained optimization instead of the user defined `optim.method`. It is possible to only set either maximum or minimum limits.

If `se.fit` is set to TRUE, standard errors of fitted values are calculated using the delta method. The standard errors of fitted values in the estimation data set are calculated as \( \text{se}_{\text{fit}} = \sqrt{G^T \Sigma G} \), where \( G \) is the gradient of a fitted value with respect to changes of parameter estimates, and \( \Sigma \) is the estimated covariance matrix of parameters (Dowd et al., 2014). The standard errors of predicted values in new data sets are calculated as \( \text{se}_{\text{pred}} = \sqrt{\text{MSE} + G^T \Sigma G} \), where \( \text{MSE} \) is the mean squared error of fitted versus observed outcomes in the original estimation data (Whitmore, 1986).

Value

`aldvmm()` returns an object of class inheriting from "aldvmm". An object of class "aldvmm" is a list containing the following objects.

- `coef` a numeric vector of parameter estimates.
- `se` a numeric vector of standard errors of parameter estimates.
- `z` a numeric vector of standardized parameter estimates.
- `p` a numeric vector of p-values of parameter estimates.
lower a numeric vector of 95% lower confidence limits of parameter estimates.
upper a numeric vector of 95% upper confidence limits of parameter estimates.
hessian a numeric matrix object with second partial derivatives of the likelihood function.
cov a numeric matrix object with covariances of parameters.
n a scalar representing the number of complete observations with no missing values that were used in the estimation.
k a scalar representing the number of components that were mixed.
gof a list including the following elements.
  ll a numeric value of the negative log-likelihood $-ll$.
  aic a numeric value of the Akaike information criterion $AIC = 2n_{par} - 2ll$.
  bic a numeric value of the Bayesian information criterion $BIC = n_{par} \times \log(n_{obs}) - 2ll$.
  mse a numeric value of the mean squared error $\sum (y - \hat{y})^2 / (n_{obs} - n_{par})$.
  mae a numeric value of the mean absolute error $\sum |y - \hat{y}| / (n_{obs} - n_{par})$.
pred a list including the following elements.
  y a numeric vector of observed outcomes in 'data'.
  yhat a numeric vector of fitted values.
  res a numeric vector of residuals.
  se.fit a numeric vector of the standard error of fitted values.
  lower.fit a numeric vector of 95% lower confidence limits of fitted values.
  upper.fit a numeric vector of 95% upper confidence limits of fitted values.
  prob a numeric vector expected values of the probabilities of group membership.
init a list including the following elements.
  est a numeric vector of initial parameter estimates.
  lo a numeric vector of lower limits of parameter estimates.
  hi a numeric vector of upper limits of parameter estimates.
formula an object of class stats::formula supplied to argument 'formula'.
psi a numeric vector with the minimum and maximum utility below 1 in 'data'.
dist a character value indicating the used distribution.
label a list including the following elements.
  lcoef a character vector of labels for objects including results on distributions (default "beta") and the probabilities of component membership (default "delta").
  lcpar a character vector of labels for objects including constant distribution parameters (default "sigma" for dist = "normal").
  lcmp a character value of the label for objects including results on different components (default "Comp")
  lvar a list including 2 character vectors of covariate names for model parameters of distributions ("beta") and the multinomial logit ("delta").
optim.method a character value of the used \texttt{optimr::optimr()} method.

The generic function \texttt{base::summary()} can be used to obtain or print a summary of the results. The generic function \texttt{stats::predict()} can be used to obtain predicted values and standard errors of predictions in new data.

References


Examples

```r
data(utility)
fit <- aldvmm(eq5d ~ age + female | 1,
data = utility,
psi = c(0.883, -0.594),
ncmp = 2)

summary(fit)

yhat <- predict(fit,
    newdata = utility)
```

---

\texttt{aldvmm.check} \hspace{1cm} \textit{Checking the Validity of Objects Supplied to aldvmm()}

**Description**

\texttt{aldvmm.check()} runs validity checks of objects supplied to \texttt{aldvmm()}.

**Usage**

```r
aldvmm.check(
    formula,
data,psi,ncmp,dist,lcoef,lcpar,lcmp,
)```

Arguments

**formula**  
an object of class "formula" with a symbolic description of the model to be fitted. The model formula takes the form \( y \sim x_1 + x_2 | x_1 + x_4 \), where the | delimiter separates the model for expected values of normal distributions (left) and the multinomial logit model of probabilities of component membership (right).

**data**  
a data frame, list or environment (or object coercible to a data frame by base::as.data.frame()) including data on outcomes and explanatory variables in 'formula'.

**psi**  
a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. \((-0.594, 0.883)\)). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in 'psi' does not matter.

**ncmp**  
a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.

**dist**  
an optional character value of the distribution used in the finite mixture. In this release, only the normal distribution is available, and the default value is set to "normal".

**lcoef**  
a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").

**lcpar**  
a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.

**lcmp**  
a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvmm().

**init.method**  
an optional character value indicating the method for obtaining initial values. The following values are available: "zero", "random", "constant" and "sann". The default value is "zero".

**optim.method**  
an optional character value of one of the following optimr::optimr() methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "Nelder-Mead". The method "L-BFGS-B" is
used when lower and/or upper constraints are set using 'init.lo' and 'init.hi'.
The method "nlm" cannot be used in the 'aldvmm' package.

optim.grad an optional logical value indicating if a numerical gradient should be used in optimr::optimr() methods that can use this information. The default value is TRUE. If 'optim.grad' is set to FALSE, a finite difference approximation is used.

init.est an optional numeric vector of user-defined initial values. User-defined initial values override the 'init.method' argument. Initial values have to follow the same order as parameter estimates in the return value 'par'.

init.lo an optional numeric vector of user-defined lower limits for constrained optimization. When 'init.lo' is not NULL, the optimization method "L-BFGS-B" is used. Lower limits of parameters have to follow the same order as parameter estimates in the return value 'par'.

init.hi an optional numeric vector of user-defined upper limits for constrained optimization. When 'init.hi' is not NULL, the optimization method "L-BFGS-B" is used. Upper limits of parameters have to follow the same order as parameter estimates in the return value 'par'.

optim.control an optional list of optimr::optimr() control parameters.

se.fit an optional logical value indicating whether standard errors of fitted values are calculated. The default value is FALSE.

level a numeric value of the significance level for confidence bands of fitted values. The default value is 0.95.

Details

aldvmm.check() checks the validity of arguments of aldvmm().

Value

aldvmm.check returns warnings or stops the execution of aldvmm() if validity checks fail.

description

aldvmm.cv() performs a numerical approximation of the covariance matrix of parameter estimates.

Usage

aldvmm.cv(ll, par, X, y, dist, psi, ncmp, lcoef, lpar, lcmp, optim.method)
Arguments

- **ll**: a function returning the negative log-likelihood of the adjusted limited dependent variable mixture model as a scalar result (aldvmm.ll()).
- **par**: a named numeric vector of parameter values.
- **X**: a list of design matrices returned by aldvmm.mm(). 'X' is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.
- **y**: a numeric vector of observed outcomes from complete observations in 'data' supplied to aldvmm().
- **dist**: an optional character value of the distribution used in the finite mixture. In this release, only the normal distribution is available, and the default value is set to "normal".
- **psi**: a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. c(-0.594,0.883)). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in 'psi' does not matter.
- **ncmp**: a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.
- **lcoef**: a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").
- **lcpar**: a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.
- **lcmp**: a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvmm().
- **optim.method**: an optional character value of one of the following optimr::optimr() methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "Nelder-Mead". The method "L-BFGS-B" is used when lower and/or upper constraints are set using 'init.lo' and 'init.hi'. The method "nlm" cannot be used in the 'aldvmm' package.

Details

aldvmm.cv() uses numDeriv::hessian() to calculate the hessian matrix of the log-likelihood function supplied to 'll' at parameter values supplied to 'par'.

Value

aldvmm.cv returns a list with the following objects.

- **hessian**: a numeric matrix with second-order partial derivatives of the likelihood function 'll'.

Creating Names of Parameter Vectors

aldvmm.getnames() creates names of parameter vectors used in aldvmm(). The order of the elements in 'lcoef' and 'lcpar' determines the order of parameters and the structure of summary tables returned by summary.aldvmm().

Usage

```r
aldvmm.getnames(X, names, lcoef, lcpar, lcmp, ncmp)
```

Arguments

- `X`: a list of design matrices returned by aldvmm.mm(). 'X' is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.
- `names`: a character vector of names of considered elements (distributions "beta", multinomial logit "delta" or constant distribution parameters, i.e. "lnsigma"). The elements in 'names' are combined with covariate names in 'X' and component labels in 'lcmp' to create a vector of names of parameter vectors.
- `lcoef`: a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").
- `lcpar`: a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.
- `lcmp`: a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvmm().
- `ncmp`: a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.

Value

a character vector of names of parameter vectors used in aldvmm().
aldvmm.getpar  Extracting Parameters from Parameter Vector into Nested List.

Description

aldvmm.getpar() extracts parameters from parameter vectors into nested lists.

Usage

aldvmm.getpar(par, lcoef, lcmp, lcpar, ncmp)

Arguments

par  a named numeric vector of parameter values.

lcoef  a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").

lcmp  a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvmm().

lcpar  a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.

ncmp  a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.

Details

aldvmm.getpar() identifies parameters that belong to coefficients of component distributions (label "beta"), coefficients of the multinomial logit model of probabilities of component membership (label "delta"), constant parameters of component distributions (label "lnsigma" for dist="normal"), and 1:K components (labels "Comp1" ... "CompK") based on parameter names generated by aldvmm.getnames().

Value

a named nested list with parameter vectors for "beta", "delta" and "lnsigma" within all 1:K components. The names of the list correspond to the labels in 'lcoef', 'lcpar' and 'lcmp'.

Calculating Goodness of Fit Measures

Description

`aldvmm.gof()` calculates residual- and likelihood-based goodness of fit measures.

Usage

`aldvmm.gof(res, par, ll)`

Arguments

- `res` a numeric vector of residuals of all observations in the estimation data.
- `par` a named numeric vector of parameter estimates.
- `ll` a numeric value of the log-likelihood.

Details

`aldvmm.gof()` calculates mean squared errors as \[ MSE = \frac{1}{n-k} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \]
and mean absolute errors as \[ MAE = \frac{1}{n-k} \sum_{i=1}^{n} |y_i - \hat{y}_i|, \]
where \( y_i \) denotes observed outcomes, \( \hat{y}_i \) denotes fitted values, \( n \) denotes the sample size, and \( k \) denotes the number of parameters. The Akaike information criterion is calculated as \( 2k - 2ll \), and the Bayesian information criterion is calculated as \( k \log(n) - 2ll \), where \( ll \) denotes the log-likelihood.

Value

`aldvmm.gof()` returns a list including the following objects.

- `mse` a numeric value of the mean squared error of observed versus fitted outcomes.
- `mae` a numeric value of the mean absolute error of observed versus fitted outcomes.
- `ll` a numeric value of the negative log-likelihood.
- `aic` a numeric value of the Akaike information criterion.
- `bic` a numeric value of the Bayesian information criterion.
Calculating Numeric Gradients of the Negative Log-Likelihood

**Description**

`aldvmm.gr()` calculates numerical gradients of the negative log-likelihood returned by `aldvmm.ll()` with respect to parameter values in `par`.

**Usage**

```r
aldvmm.gr(par, X, y, psi, dist, ncmp, lcoef, lcmp, lcpar, optim.method)
```

**Arguments**

- `par`: a named numeric vector of parameter values.
- `X`: a list of design matrices returned by `aldvmm.mm()`. ‘X’ is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.
- `y`: a numeric vector of observed outcomes from complete observations in 'data' supplied to `aldvmm()`.
- `psi`: a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. c(-0.594, 0.883)). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in 'psi' does not matter.
- `dist`: an optional character value of the distribution used in the finite mixture. In this release, only the normal distribution is available, and the default value is set to "normal".
- `ncmp`: a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.
- `lcoef`: a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").
- `lcmp`: a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by `summary.aldvmm()`.
- `lcpar`: a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.
- `optim.method`: an optional character value of one of the following `optimr::optimr()` methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "Nelder-Mead". The method "L-BFGS-B" is used when lower and/or upper constraints are set using 'init.lo' and 'init.hi'. The method "nlm" cannot be used in the 'aldvmm' package.
Details

aldvmm.gr() uses numDeriv::grad() to perform numerical approximation of gradients of the negative log-likelihood returned by aldvmm.ll().

Value

a named numeric vector of first derivatives of the negative log-likelihood of the data with respect to parameters in 'par'.

aldvmm.init

Creating Initial Values

Description

aldvmm.init() creates initial values for the minimization of the negative log-likelihood returned by aldvmm.ll() using optimr::optimr().

Usage

aldvmm.init(
  X,
  y,
  psi,
  ncmp,
  dist,
  init.method,
  init.est,
  init.lo,
  init.hi,
  optim.method,
  optim.control = list(),
  optim.grad = TRUE,
  lcoef,
  lcpar,
  lcmp
)

Arguments

X      a list of design matrices returned by aldvmm.mm(). 'X' is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.
y      a numeric vector of observed outcomes from complete observations in 'data' supplied to aldvmm().
psi

a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. c(-0.594, 0.883)). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in 'psi' does not matter.

ncmp

a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.

dist

an optional character value of the distribution used in the finite mixture. In this release, only the normal distribution is available, and the default value is set to "normal".

init.method

an optional character value indicating the method for obtaining initial values. The following values are available: "zero", "random", "constant" and "sann". The default value is "zero".

init.est

an optional numeric vector of user-defined initial values. User-defined initial values override the 'init.method' argument. Initial values have to follow the same order as parameter estimates in the return value 'par'.

init.lo

an optional numeric vector of user-defined lower limits for constrained optimization. When 'init.lo' is not NULL, the optimization method "L-BFGS-B" is used. Lower limits of parameters have to follow the same order as parameter estimates in the return value 'par'.

init.hi

an optional numeric vector of user-defined upper limits for constrained optimization. When 'init.hi' is not NULL, the optimization method "L-BFGS-B" is used. Upper limits of parameters have to follow the same order as parameter estimates in the return value 'par'.

optim.method

an optional character value of one of the following optimr::optimr() methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hjn". The default method is "Nelder-Mead". The method "L-BFGS-B" is used when lower and/or upper constraints are set using 'init.lo' and 'init.hi'. The method "nlm" cannot be used in the 'aldvmm' package.

optim.control

an optional list of optimr::optimr() control parameters.

optim.grad

an optional logical value indicating if a numerical gradient should be used in optimr::optimr() methods that can use this information. The default value is TRUE. If 'optim.grad' is set to FALSE, a finite difference approximation is used.

lcoef

a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").

lcpar

a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.

lcmp

a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvmm().
Details

'init.method' accepts four methods for generating initial values: "zero", "random", "constant", 
"sann". The method "zero" sets initial values of all parameters to 0. The method "random" draws 
random starting values from a standard normal distribution. The method "constant" estimates 
a constant-only model and uses estimates as initial values for intercepts and constant distribution 
parameters and 0 for all other parameters. The method "sann" estimates the full model using the 
simulated annealing optimization method and uses all parameter estimates as initial values. When 
user-specified initial values are supplied in 'init.est', the argument 'init.method' is ignored.

By default, aldvmm() performs unconstrained optimization with upper and lower limits at -Inf and 
Inf. When user-defined lower and upper limits are supplied to 'init.lo' and/or 'init-hi', these 
default limits are replaced with the user-specified values, and the method "L-BFGS-B" is used for 
box-constrained optimization instead of the user defined 'optim.method'. It is possible to only set 
either maximum or minimum limits.

Value

aldvmm.init() returns a list with the following objects.

est     a numeric vector of initial values of parameters.
lo      a numeric vector of lower limits of parameters.
hi      a numeric vector of upper limits of parameters.

aldvmm.ll

Calculating the Negative Log-Likelihood of the Adjusted Limited Dependent Variable Mixture Model

Description

aldvmm.ll() calculates the negative log-likelihood of 'data' supplied to aldvmm() at the parameter 
values in 'par'.

Usage

aldvmm.ll(par, X, y, psi, ncmp, dist, lcoef, lpar, lcmp, optim.method)

Arguments

par       a named numeric vector of parameter values.
X         a list of design matrices returned by aldvmm.mm(). 'X' is of length 2 and in-
           cludes a design matrix for the model of component distributions and a design 
           matrix for the model of probabilities of group membership.
y         a numeric vector of observed outcomes from complete observations in 'data' 
           supplied to aldvmm().
psi a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. c(-0.594, 0.883)). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in 'psi' does not matter.

ncmp a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.

dist an optional character value of the distribution used in the finite mixture. In this release, only the normal distribution is available, and the default value is set to "normal".

lcoef a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").

lcpar a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.

lcmp a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvmm().

optim.method an optional character value of one of the following optimr::optimr() methods: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "nlminb", "Rcgmin", "Rvmmin" and "hn". The default method is "Nelder-Mead". The method "L-BFGS-B" is used when lower and/or upper constraints are set using 'init.lo' and 'init.hi'. The method "nlm" cannot be used in the 'aldvmm' package.

Details

aldvmm.ll() calculates the negative log-likelihood of the adjusted limited dependent variable mixture model using the likelihood function published in Hernandez Alava and Wailoo (2015). Constant distribution parameters that need to be non-negative (i.e. the standard deviations of normal distributions) enter the likelihood function as log-transformed values.

As the "L-BFGS-B" and "Rcgmin" methods in optimr::optimr() fail if they encounter infinite values, the log-likelihood function takes the value -1e+20 if it is infinite during these algorithms.

The names of the parameter vector supplied to 'par' must be generated using aldvmm.getnames() because they will be inherited by return values of other functions in the package 'aldvmm'. The names will also be used in the extraction of parameters from parameter vectors into nested lists using aldvmm.getpar().

Value

a scalar of the negative log-likelihood of the data at parameter values in 'par'.

References

aldvmm.mm

Creating Design Matrices

Description

aldvmm.mm() creates two design matrices, one of the model of component distributions ("beta") and one of the model of probabilities of component membership ("delta").

Usage

aldvmm.mm(data, formula, ncmp, lcoef)

Arguments

data a data frame, list or environment (or object coercible to a data frame by base::as.data.frame()) including data on outcomes and explanatory variables in 'formula'.

formula an object of class "formula" with a symbolic description of the model to be fitted. The model formula takes the form y ~ x1 + x2 | x1 + x4, where the | delimiter separates the model for expected values of normal distributions (left) and the multinomial logit model of probabilities of component membership (right).

ncmp a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.

lcoef a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").

Details

aldvmm.mm() uses stats::model.matrix() to create design matrices for models of component distributions ("beta") and probabilities of component membership ("delta") based on 'formula' supplied to aldvmm.ll(). The design matrix for probabilities of group membership is only created if more than one components are specified in 'ncmp'.

Value

a named list of numeric matrices.

beta a numeric design matrix for the model of component distributions.

delta a numeric design matrix of the multinomial logit model of probabilities of component membership.
aldvmm.pred

*Predicting Expected Values from Adjusted Limited Dependent Variable Mixture Models*

**Description**

`aldvmm.pred()` makes predictions of observations in design matrices in 'X' using parameter estimates returned by `aldvmm()`.

**Usage**

`aldvmm.pred(par, X, y = NULL, psi, ncmp, dist, lcoef, lpar, lcmp)`

**Arguments**

- `par`: a named numeric vector of parameter values.
- `X`: a list of design matrices returned by `aldvmm.mm()`. 'X' is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.
- `y`: a numeric vector of observed outcomes from complete observations in 'data' supplied to `aldvmm()`.
- `psi`: a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. `c(-0.594, 0.883)`). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in 'psi' does not matter.
- `ncmp`: a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.
- `dist`: an optional character value of the distribution used in the finite mixture. In this release, only the normal distribution is available, and the default value is set to "normal".
- `lcoef`: a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").
- `lpar`: a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lpar' depends on the distribution supplied to 'dist'.
- `lcmp`: a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by `summary.aldvmm()`.
aldvmm.sefit

Details

aldvmm.pred() calculates expected values for observations in design matrices in 'X' using the expected value function published in Hernandez Alava and Wailoo (2015). Constant distribution parameters that need to be non-negative (i.e. standard deviations of normal distributions) enter the expected value function as log-transformed values.

Value

A named numeric vector of predicted outcomes. The names of the elements in the vector are identical to the row names of design matrices in 'X'.

Description

aldvmm.sefit() calculates standard errors of fitted and predicted outcomes using the delta method.

Usage

aldvmm.sefit(
  par,
  yhat,
  X,
  type,
  formula,
  psi,
  cv,
  mse = NA,
  ncmp,
  dist,
  level,
  lcoef,
  lcmp,
  lcpar
)

Arguments

par a named numeric vector of parameter values.

yhat a numeric vector of predicted outcomes returned by aldvmm.pred().

X a list of design matrices returned by aldvmm.mm(). 'X' is of length 2 and includes a design matrix for the model of component distributions and a design matrix for the model of probabilities of group membership.

type a character value of either 'fit' or 'pred' indicating whether the standard error of the fit ('fit') or the standard error of predictions in new data ('pred') are calculated.
an object of class "formula" with a symbolic description of the model to be fitted. The model formula takes the form y ~ x1 + x2 | x1 + x4, where the | delimiter separates the model for expected values of normal distributions (left) and the multinomial logit model of probabilities of component membership (right).

a numeric vector of minimum and maximum possible utility values smaller than or equal to 1 (e.g. c(-0.594, 0.883)). The potential gap between the maximum value and 1 represents an area with zero density in the value set from which utilities were obtained. The order of the minimum and maximum limits in 'psi' does not matter.

a numeric matrix with covariances/variances of parameter estimates returned by aldvm.m.cv().

a numeric value of the mean squared error of observed versus predicted outcomes \( \sum (y - \hat{y})^2 / (n_{obs} - n_{par}) \) for all observations in model matrices 'X' supplied to aldvm.m.11().

a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.

an optional character value of the distribution used in the finite mixture. In this release, only the normal distribution is available, and the default value is set to "normal".

a numeric value of the significance level for confidence bands of fitted values. The default value is 0.95.

a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").

a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvm.m().

a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.

Details

aldvm.m.sefit() calculates standard errors of fitted values using the delta method. Standard errors of fitted values in the estimation data set are calculated as \( se_{fit} = \sqrt{G'} \Sigma G \), where \( G \) is the gradient of a fitted value with respect to changes of parameter estimates, and \( \Sigma \) is the estimated covariance matrix of parameters (Dowd et al., 2014). Standard errors of predicted values in new data sets are calculated as \( se_{pred} = \sqrt{MSE + G' \Sigma G} \), where \( MSE \) is the mean squared error of fitted versus observed outcomes in the original estimation data (Whitmore, 1986). The gradients of fitted values with respect to parameter estimates are approximated numerically using numDeriv::jacobian().

Value

a named numeric vector of standard errors of fitted or predicted outcomes. The names of the elements in the vector are identical to the row names of design matrices in 'X'.
References


aldvmm.sum

Creating Summary Table

Description
aldvmm.sum() creates a summary table of regression results.

Usage
aldvmm.sum(
est,
se,
z,
p,
lower,
upper,
n,
value,
aic,
bic,
ncmp,
lcoef,
lpar,
lcmp,
lvar,
digits = 3
)

Arguments

est a named numeric vector of point estimates.
se a named numeric vector of standard errors of parameters returned by aldvmm.cv().
z a named numeric vector of standardized coefficients of parameters returned by aldvmm.cv().
p a named numeric vector of p-values of parameters returned by aldvmm.cv().
lower a named numeric vector of 95% lower limits of parameters returned by aldvmm.cv().
upper a named numeric vector of 95% upper limits of parameters returned by aldvmm.cv().
a numeric value of the number of complete observations in 'data' supplied to aldvmm().

value
a numeric value of the negative log-likelihood returned by aldvmm.ll().

aic
a numeric value of the Akaike information criterion (AIC) returned by aldvmm.gof().

bic
a numeric value of the Bayesian information criterion (BIC) returned by aldvmm.gof().

ncmp
a numeric value of the number of components that are mixed. The default value is 2. A value of 1 represents a tobit model with a gap between 1 and the maximum value in 'psi'.

lcoef
a character vector of length 2 with labels of objects including regression coefficients of component distributions (default "beta") and coefficients of probabilities of component membership (default "delta").

lcpar
a character vector with the labels of objects including constant parameters of component distributions (e.g. the standard deviation of the normal distribution). The length of 'lcpar' depends on the distribution supplied to 'dist'.

lcmp
a character value representing a stub (default "Comp") for labeling objects including regression coefficients in different components (e.g. "Comp1", "Comp2", ...). This label is also used in summary tables returned by summary.aldvmm().

lvar
a named list of character vectors with column names of design matrices returned by aldvmm.mm().

digits
a numeric value of the number of digits in the reporting table.

Value

a data.frame object with a summary table of regression results.

Description

The generic function stats::predict() calls aldvmm.pred() to predict outcomes and their standard errors in new data using aldvmm.pred() and aldvmm.sefit().

Usage

```r
## S3 method for class 'aldvmm'
predict(object, newdata, se.fit = FALSE, type = "pred", level = 0.95, ...)
```
summary.aldvmm

Arguments

**object**
an object inheriting from class 'aldvmm'.

**newdata**
a data frame, list or environment (or object coercible to a data frame by `base::as.data.frame()`) including explanatory variables for prediction.

**se.fit**
an optional logical value indicating whether standard errors of fitted values are calculated. The default value is `FALSE`.

**type**
a character value of either 'fit' or 'pred' indicating whether the standard error of the fit ('fit') or the standard error of predictions in new data ('pred') are calculated.

**level**
a numeric value of the significance level for confidence bands of fitted values. The default value is 0.95.

**...**
further arguments passed to or from other methods.

Value

a named list of numeric vectors of predicted outcomes, standard errors and confidence or prediction intervals.

---

**summary.aldvmm**

*Summarizing Adjusted Limited Dependent Variable Mixture Model Fits*

Description

The generic function `base::summary()` calls `aldvmm.sum()` to print the summary table data frame returned by `aldvmm.sum()`.

Usage

```
## S3 method for class 'aldvmm'
summary(object, ...)
```

Arguments

**object**
an object inheriting from class 'aldvmm'.

**...**
further arguments passed to or from other methods.
Simulated Example Data of Health State Utilities.

**Description**

utility is a simulated data frame including health state utilities and patients’ age and sex.

**Usage**

utility

**Format**

A data frame with 200 rows and 3 variables:

- **utility**: a utility value \([-0.594, 0.883]\], 1.
- **age**: Age in years.
- **female**: Indicator of female sex.

**Examples**

```r
set.seed(101010101)
utility <- data.frame(female = rbinom(size = 1, n = 200, p = 0.6))
utility[, 'age'] <- stats::rnorm(n = 200, mean = 50 + utility$female*10, sd = 15)
utility[1:50, 'eq5d'] <- stats::rnorm(n = 50, mean = 0 - 0.1 * utility[1:50, 'female'] + 0.0005 * utility[1:50, 'age'], sd = 0.1)
utility[51:200, 'eq5d'] <- stats::rnorm(n = 150, mean = 0.5 + 0.1 * utility[51:200, 'female'] + 0.0001*utility[51:200, 'age'], sd = 0.2)
utility[utility$eq5d<(-0.594), 'eq5d'] <- -0.594
utility[utility$eq5d>0.883, 'eq5d'] <- 1
hist(utility$eq5d, breaks = 50)
```
Index

* datasets
  · utility, 26

aldvmm, 3
aldvmm-package, 2
aldvmm.check, 7
aldvmm.cv, 9
aldvmm.getnames, 11
aldvmm.getpar, 12
aldvmm.gof, 13
aldvmm.gr, 14
aldvmm.init, 15
aldvmm.ll, 17
aldvmm.mm, 19
aldvmm.pred, 20
aldvmm.sefit, 21
aldvmm.sum, 23

predict.aldvmm, 24

summary.aldvmm, 25

utility, 26