Package ‘crov’

May 8, 2026

Type Package

Title Constrained Regression Model for an Ordinal Response and Ordinal Predictors

Version 0.3.0

Author Javier Espinosa <javier.espinosa.b@usach.cl>

Maintainer Javier Espinosa <javier.espinosa.b@usach.cl>

Description Fits a constrained regression model for an ordinal response with ordinal predictors and possibly others, Espinosa and Hennig (2019) <DOI:10.1007/s11222-018-9842-2>. The parameter estimates associated with an ordinal predictor are constrained to be monotonic. If a monotonicity direction (isotonic or antitonic) is not specified for an ordinal predictor by the user, then one of the available methods will either establish it or drop the monotonicity assumption. Two monotonicity tests are also available to test the null hypothesis of monotonicity over a set of parameters associated with an ordinal predictor.

License GPL-2

Encoding UTF-8

LazyData TRUE

Imports VGAM (>= 1.0-5), gtools (>= 3.5.0), stats (>= 3.4.3)

RoxygenNote 7.2.3

NeedsCompilation no

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2023-08-25 13:50:09 UTC

Contents

confRegCCR 2
confRegUCRandUCCR 3
crovData 5
mdcp 5
monoTestBonf 8
monoTestConfReg 9
plotCMLE 10

Index 12

`confRegCCR` *Parameter Vector in Confidence Region CCR*

Description

Determines whether a parameter vector is in the confidence region CCR, according to the definitions in Espinosa and Hennig (2023) <https://doi.org/10.48550/arXiv.2107.04946>.

Usage

```
confRegCCR(
  CMLE = NULL,
  paramVals = NULL,
  paramIDs = NULL,
  SignifLevel = 0.05,
  df,
  matY,
  matX
)
```

Arguments

`CMLE`	A vector with the constrained maximum likelihood estimates.
`paramVals`	A vector with the parameter values for which it is needed to assess whether it is part of the confidence region or not. The order of the parameters must be the same as the one of `CMLE`. As in Espinosa and Hennig (2023), the parameter vector contains the parameters of interest, beta_0r, and the remaining ones are the constrained MLEs given beta_0r.
`paramIDs`	A vector indicating the positions of the parameter values of beta_0r in `paramVals`, which are those of interest, usually the ones corresponding to some ordinal predictor. For instance, `paramIDs=7:11` indicates that the 7th to the 11th parameter values in `paramVals` are the ones of interest and correspond to some ordinal predictor.
`SignifLevel`	A decimal number indicating the significant level. Usually, 0.05.
`df`	Degrees of freedom to be used.
`matY`	matY resulting from mdcp().
`matX`	matX resulting from mdcp().

Value

`confRegions`: Data frame with columns: `CMLE_logLik`=log-likelihood of the constrained model, `param_logLik`=log-likelihood of the model using `paramVals`, `monotonicBeta0`=logical value, `TRUE` if the set of parameters of `paramVals` indicated by `paramIDs` are monotonic, `df`=degrees of freedom used to calculate the critical value, `StatCCR`=value of the statistic used for `CCR`, `CritValue`=critical value, chi-squared with `df` and `1-SignifLevel`, `SignifLevel`=significance level used to calculate the critical value, `inCCR`=logical value, `TRUE` if `paramVals` belongs to the confidence region `CCR`,

References

Espinosa, J., and Hennig, C. "Inference for the proportional odds cumulative logit model with monotonicity constraints for ordinal predictors and ordinal response." Arxiv (2023). <https://doi.org/10.48550/arXiv.2107.04

See Also

confRegUCRandUCCR, mdcp, monoTestBonf, monoTestConfReg, plotCMLE, vlgm.

Examples

```
resAux <- mdcp(QoL ~ EduLevel + Age + IncQuint + Gender + Health, data = crovData)
plotCMLE(resAux)
myVector <- resAux$estimates
myVectorID <- 10:12
myVector[myVectorID]

# non-monotonic beta_{0r}
myVector[myVectorID] <- seq(0.195,0.185,length.out=3)
confRegCCR(CMLE=resAux$estimates, paramVals=myVector, paramIDs=myVectorID,SignifLevel=0.05, df=3,
matY= resAux$matY, matX= resAux$matX)

# monotonic beta_{0r} and paramVals in CCR
myVector[myVectorID] <- seq(0.048,0.049,length.out=3)
confRegCCR(CMLE=resAux$estimates, paramVals=myVector, paramIDs=myVectorID,SignifLevel=0.05, df=3,
matY= resAux$matY, matX= resAux$matX)

# monotonic beta_{0r} and paramVals out of CCR
myVector[myVectorID] <- seq(0.047,0.048,length.out=3)
confRegCCR(CMLE=resAux$estimates, paramVals=myVector, paramIDs=myVectorID,SignifLevel=0.05, df=3,
matY= resAux$matY, matX= resAux$matX)
```

---

confRegUCRandUCCR *Parameter Vector in Confidence Regions UCR and/or UCCR*

Description

Determines whether a parameter vector is in the confidence region UCR and/or UCCR, according to the definitions in Espinosa and Hennig (2023) <https://doi.org/10.48550/arXiv.2107.04946>.

Usage

```
confRegUCRandUCCR(
  UMLE = NULL,
  paramVals = NULL,
  paramIDs = NULL,
  SignifLevel = 0.05,
  df,
  matY,
  matX
)
```

Arguments

UMLE	A vector with the unconstrained maximum likelihood estimates.
paramVals	A vector with the parameter values for which it is needed to assess whether it is part of one of the confidence regions or not. The order of the parameters must be the same as the one of UMLE. As in Espinosa and Hennig (2023), the parameter vector contains the parameters of interest, beta_0r, and the remaining ones are the unconstrained MLEs given beta_0r.
paramIDs	A vector indicating the positions of the parameter values of beta_0r in paramVals, which are those of interest, usually the ones corresponding to some ordinal predictor. For instance, paramIDs=7:11 indicates that the 7th to the 11th parameter values in paramVals are the ones of interest and correspond to some ordinal predictor.
SignifLevel	A decimal number indicating the significant level. Usually, 0.05.
df	Degrees of freedom to be used.
matY	matY resulting from mdcp().
matX	matX resulting from mdcp().

Value

confRegions: Data frame with columns: UMLE_logLik=log-likelihood of the unconstrained model, param_logLik=log-likelihood of the model using paramVals, monotonicBeta0=logical value, TRUE if the set of parameters of paramVals indicated by paramIDs are monotonic, df=degrees of freedom used to calculate the critical value, StatUCR=value of the statistic used for UCR, StatUCCR=value of the statistic used for UCCR, CritValue=critical value, chi-squared with df and 1-SignifLevel, SignifLevel=significance level used to calculate the critical value, inUCR=logical value, TRUE if paramVals belongs to the confidence region UCR, inUCCR=logical value, TRUE if paramVals belongs to the confidence region UCCR,

References

Espinosa, J., and Hennig, C. "Inference for the proportional odds cumulative logit model with monotonicity constraints for ordinal predictors and ordinal response." Arxiv (2023). <https://doi.org/10.48550/arXiv.2107.04

See Also

confRegCCR, mdcp, monoTestBonf, monoTestConfReg, plotCMLE, vlgm.

Examples

```
resAux <- mdcp(QoL ~ EduLevel + Age + IncQuint + Gender + Health, data = crovData)
plotCMLE(resAux)
myVector <- resAux$estimates
myVectorID <- 10:12
myVector[myVectorID]

# non-monotonic beta_{0r}, paramVals in UCR but not in UCCR
myVector[myVectorID] <- seq(0.195,0.185,length.out=3)
confRegUCRandUCCR(UMLE=resAux$UMLE, paramVals=myVector, paramIDs=myVectorID,SignifLevel=0.05, df=3,
```

```
matY= resAux$matY, matX= resAux$matX)

# monotonic beta_{0r}, paramVals in UCR and UCCR
myVector[myVectorID] <- seq(0.073,0.074,length.out=3)
confRegUCRandUCCR(UMLE=resAux$UMLE, paramVals=myVector, paramIDs=myVectorID,SignifLevel=0.05, df=3,
matY= resAux$matY, matX= resAux$matX)

# monotonic beta_{0r}, paramVals out of UCR and UCCR
myVector[myVectorID] <- seq(0.072,0.073,length.out=3)
confRegUCRandUCCR(UMLE=resAux$UMLE, paramVals=myVector, paramIDs=myVectorID,SignifLevel=0.05, df=3,
matY= resAux$matY, matX= resAux$matX)
```

`crovData` *Real data example*

Description

Uses real data

Usage

```
crovData
```

Format

Data frame with 9 variables

`mdcp` *Monotonicity Direction Classification (MDC) procedure*

Description

Fits a constrained regression model for an ordinal response with ordinal predictors and possibly others, Espinosa and Hennig (2019) <DOI:10.1007/s11222-018-9842-2>. The parameter estimates associated with an ordinal predictor are constrained to be monotonic. If a monotonicity direction (isotonic or antitonic) is not specified for an ordinal predictor (OP) by the user, then a constrained method to be indicated in the option `method` establishes it or the approach that tries all possible combinations of monotonicity directions an the chooses the one with maximum likelihood.

Usage

```
mdcp(
  formula,
  data = NULL,
  tryAllMonoDir = FALSE,
  monoDir = NULL,
  CLS1 = 0.95,
  TLBS2 = 0.85,
  TLNS2 = 0.999,
  StepSizeCLS2 = 1e-04,
  method = NULL,
  monoTestSignLevel = 0.05,
  reltol = 1e-05,
  mu = 1e-04,
  outer.eps = 1e-05
)
```

Arguments

`formula` A `formula` to be fitted with ordinal response, one or more ordinal predictors, and possibly one or more other predictors. For ordinal response and ordinal predictors use ordered factors.

`data` A data.frame, list or environment (or object coercible by `as.data.frame` to a data.frame), containing the variables in `formula`. Neither a matrix nor an array will be accepted.

`tryAllMonoDir` A logical value that indicates whether one model should be fitted for each one of the possible combinations of monotonicity directions of the ordinal predictors' effects. Use `TRUE` if no constrained method is used in `method`.

`monoDir` Vector with monotonicity directions for the ordinal predictors to be used as constraints. Possible values for `monoDir` are `1`, `0` and `-1`. Use `1` for "isotonic" and `-1` for "antitonic". If the monotonicity direction of the prameters of an ordinal predictor has to be estimated, then use `0`, which also allows to drop the monotonicity assumption when `method` is not "MDCS3". The order of the elements in `monoDir` must be the same as the order of the ordinal predictors in the object `formula`, i.e., the j-th element of `monoDir` must correspond to the monotonicity direction of the j-th ordinal predictor in `formula`. For example, `monoDir=c(0,-1,-1,1,0)` indicates that the monotonicity direction of the effects of the first OP will be estimated; the monotonicity direction of the effects of the second and third OPs are antitonic; the monotonicity direction of the effects of the fourth OP isotonic; and the monotonicity direction of the effects of the fifth OP will also be estimated. If `tryAllMonoDir` and `monoDir` are not used (default option), the monotonicity direction classification prodecure is executed on all of the ordinal predictors to find the monotonicity directions associated to the model with the maximum log-likelihood.

`CLS1` Numerical value for the confidence level to be used in the first step of the MDC procedure. This parameter is active if `tryAllMonoDir` is not used.

`TLBS2` Numerical value for the tolerance level to be used in the second step of the MDC procedure over those ordinal predictors classified as "Both" in the first step. This parameter is active if `tryAllMonoDir` and `monoDir` are not used.

`TLNS2` Numerical value for the tolerance level to be used in the second step of the MDC procedure over those ordinal predictors classified as "None" in the first step. This parameter is active if `tryAllMonoDir` and `monoDir` are not used.

`StepSizeCLS2` Numerical value for the magnitude in which the confidence levels will be increased or decreased during the second step of the MDC procedure. This parameter is active if `tryAllMonoDir` and `monoDir` are not used.

`method` The type of constrained method to be used among `"MDCS1"`, `"MDCS2"`, `"MDCS3"`, `"CMLEbonferroni"`, `"CMLEconfReg"`, and `"CMLEfiltered"`. Default value is `"MDCS3"`. `"MDCS1"` uses the first step of the MDC procedure described in Espinosa, J. and Christian H. (2019) to drop the monotonicity constraints on those ordinal predictors (OPs) classified as "both" or "none". `"MDCS2"` uses the second step of the MDC procedure described in Espinosa, J. and Christian H. (2019) to drop the monotonicity constraints on those OPs classified as "both" or "none". `"MDCS3"` performs the three steps of the MDC procedure described in Espinosa, J. and Christian H. (2019) and does not drop monotonicity constraints on the OPs, being the most restrictive method. `"CMLEbonferroni"` tests the null hypothesis of monotonic effects for an OP as described in Espinosa, J. and Christian H. (2019) and drops the monotonicity constraints on those OPs for which the monotonicity test rejects the null hypothesis. `"CMLEconfReg"` tests the null hypothesis of monotonic effects for an OP based on the analysis of confidence regions and drops the monotonicity constraints on those OPs for which the monotonicity test rejects the null hypothesis. `"CMLEfiltered"` uses the first step of the MDC procedure described in Espinosa, J. and Christian H. (2019) to drop the monotonicity constraints on those ordinal predictors (OPs) classified as "none".

`monoTestSignLevel` Significance level used when `method` is `"CMLEbonferroni"` or `"CMLEconfReg"`. Default value 0.05.

`reltol` Passed to `constrOptim()`.

`mu` Passed to `constrOptim()`.

`outer.eps` Passed to `constrOptim()`.

Value

`MDCproc`: Data frame with the monotonicity direction classification (Isotonic, Antitonic, Both, or None) used for each ordinal predictor in each one of the steps of the MDC procedure (S1, S2 and S3), together with their individual confidence levels (CL). If `monoDir` is used, `MDCproc` shows the monotonicity directions in `monoDir`.

`estimates`: Vector of parameter estimates of the model.

`log.lik`: Value of the log-likelihood of the model.

`allModels`: Data frame with monotonicity directions, log-likelihood and parameter estimates of all models involved in the third step of the MDC procedure. If parameter `monoDir` is used, `allModels`

shows these results from the model with monotonicity directions used in `monoDir` only. If parameter `tryAllMonoDir` is used, `allModels` shows these results from all the models according to all possible combinations of monotonicity directions.

`constrOptimRes`: List with the outcomes provided by the function `constrOptim`.

`UMLE`: Vector with the parameter estimates of the unconstrained version of the model.

`UMLE_SE`: Vector with the standard errors of the unconstrained version of the model.

`q_cat_OrdPred`: Vector with the number of ordinal categories for each ordinal predictor. Values are displayed in the same order as the ordinal predictors are incorporated in `formula`.

References

Espinosa, J., and Hennig, C. "A constrained regression model for an ordinal response with ordinal predictors." Statistics and Computing 29.5 (2019): 869-890. https://doi.org/10.1007/s11222-018-9842-2.

See Also

`monoTestBonf`, `monoTestConfReg`, `confRegUCRandUCCR`, `confRegCCR`, `plotCMLE`, `constrOptim`.

Examples

```
# Ordinal predictors: EduLevel, IncQuint and Health,
mdcpExample <- mdcp(QoL ~ EduLevel + Age + IncQuint + Gender + Health, data = crovData,
CLS1 = 0.95, TLBS2 = 0.90, TLNS2 = 0.99, StepSizeCLS2 = 0.0002)
mdcpExample$MDCproc
cbind("CMLE"=mdcpExample$estimates,"UMLE"=mdcpExample$UMLE)
mdcpExample$UMLE_SE
mdcpExample$log.lik
mdcpExample$allModels[1:6]
```

`monoTestBonf` *Monotonicity test*

Description

Tests the null hypothesis of monotonicity over a set of parameters associated to an ordinal predictor, according to Espinosa and Hennig (2019) <DOI:10.1007/s11222-018-9842-2>.

Usage

```
monoTestBonf(simultAlpha = 0.05, OP_UMLE, OP_SE)
```

Arguments

simultAlpha	Numerical value for the simultaneous significance level.
OP_UMLE	Vector with the unconstrained parameter estimates of an ordinal predictor's categories represented by dummy variables in an unconstrained model for ordinal response (see vlgm).
OP_SE	Vector with the standard error of the parameters of an ordinal predictor's categories represented by dummy variables in an unconstrained model for ordinal response (see vlgm).

Value

testRes: String value with outcomes either "Reject H_0" or "Not Reject H_0".

simultAlpha: Numerical value with the simultaneous significance level.

indivAlphaA: Numerical value with the individual significance level for each confidence interval.

simultPvalue: Numerical value with the p-value associated to the simultaneous significance level.

References

Espinosa, J., and Hennig, C. "A constrained regression model for an ordinal response with ordinal predictors." Statistics and Computing 29.5 (2019): 869-890. https://doi.org/10.1007/s11222-018-9842-2.

See Also

mdcp, monoTestConfReg, plotCMLE, vlgm.

Examples

```
monoTestBonf(simultAlpha=0.05, OP_UMLE = c(-0.352177095,-0.403928770,
-0.290875028,-0.769834449), OP_SE = c(0.246638339,0.247723681,0.267577633,0.300951441))
```

---

monoTestConfReg *Monotonicity test using confidence regions*

Description

Tests the null hypothesis of monotonicity over a set of parameters associated to an ordinal predictor. The log-likelihood ratio test is used after imposing ordinal constraints on the parameter estimates of a single ordinal predictor and comparing its results against the unconstrained MLEs.

Usage

```
monoTestConfReg(formula, data = NULL, monoDir = NULL, SignifLevel = 0.05)
```

Arguments

formula	A formula to be fitted with ordinal response, one or more ordinal predictors, and possibly one or more other predictors. For ordinal response and ordinal predictors use ordered factors.
data	A data.frame, list or environment (or object coercible by as.data.frame to a data.frame), containing the variables in formula. Neither a matrix nor an array will be accepted.
monoDir	Vector with monotonicity directions for the ordinal predictors to be used as constraints. Possible values for monoDir are 1, 0 and -1. Use 1 for "isotonic", -1 for "antitonic", and 0 o test monotonicity of the prameters of an ordinal predictor. The order of the elements in monoDir must be the same as the order of the ordinal predictors in the object formula, i.e., the j-th element of monoDir must correspond to the monotonicity direction of the j-th ordinal predictor in formula. If monoDir is not used (default option), the monotonicity of all ordinal predictors' effects are tested.
SignifLevel	Numerical value for the significance level.

Value

resConfRegTest: Data frame with columns: OPName=Name of the ordinal predictor (OP), Num_Cat=Number of categories of the OP, UMLE_logLik=log-likelihood of the unconstrained model, CMLE_logLik=log-likelihood of the constrained model using mdcp assuming monotonicity for each OP, degreesOfFreedom=degrees of freedom used in the hypothesis test, Statistic=value of the statistic, CritValue=critical value resulting from the statistic, SignifLevel=significance level used in the test, P.Value=p-value, RejectMonotonicity=TRUE if the null hypothesis of monotonicity is rejected, FALSE otherwise.

See Also

mdcp, monoTestBonf, confRegUCRandUCCR, confRegCCR, plotCMLE, vlgm.

Examples

```
# Ordinal predictors: EduLevel, IncQuint and Health
  monoTestConfRegExample <- monoTestConfReg(QoL ~ EduLevel + Age + IncQuint + Gender +
  Health, data = crovData, monoDir=c(0,-1,-1), SignifLevel = 0.05)
  monoTestConfRegExample$resConfRegTest
```

---

plotCMLE — *Plot unconstrained and constrained proportional odds logit model*

Description

Uses the results of function mdcp to produce a plot for the Maximum Likelihood Estimators of the parameters of both the unconstrained and constrained proportional odds logit models (UMLE and CMLE correspondingly). The UMLE includes confidence intervals. Parameter estimates of ordinal predictors are graphically linked with segments.

Usage

```
plotCMLE(
  mdcpResult = NULL,
  SignifLevel = 0.05,
  xposLegend = NULL,
  yposLegend = NULL,
  xcex.axis = 0.8,
  ycex.axis = 0.8,
  cexLegend = 1,
  methodName = "Not indicated"
)
```

Arguments

`mdcpResult`	An object of class `list` storing the results of function `mdcp`, which fits both the unconstrained and constrained proportional odds logit models.
`SignifLevel`	Significance level to be used when constructing the confidence intervals for each parameter of the unconstrained proportional odds logit model. Default value 0.05.
`xposLegend`	Position of legend on the x-axis. If `xposLegend` or `yposLegend` are not used, then the legend is located using `topleft` option.
`yposLegend`	Position of legend on the y-axis. If `xposLegend` or `yposLegend` are not used, then the legend is located using `topleft` option.
`xcex.axis`	Size of `cex.axis` for the x-axis. Default value is 0.8.
`ycex.axis`	Size of `cex.axis` for the y-axis. Default value is 0.8.
`cexLegend`	Size of legend text to be assigned to `cex`. Default value is 1.
`methodName`	Method name to be used in the main title of the plot.

Value

Plot.

See Also

`monoTestBonf`, `monoTestConfReg`, `monoTestBonf`, `constrOptim`.

Examples

```
# Ordinal predictors: EduLevel, IncQuint, Health,
# Overcrowd, and NumChildren
mdcpExample <- mdcp(QoL ~ EduLevel + Age + IncQuint + Gender + Health, data = crovData,
CLS1 = 0.95, TLBS2 = 0.90, TLNS2 = 0.99, StepSizeCLS2 = 0.0002)
plotCMLE(mdcpResult=mdcpExample,SignifLevel=0.05,xposLegend=14, yposLegend=4.8,
cexLegend=0.8, method="MDCS3")
```

Index

∗ **datasets**
    crovData, 5

as.data.frame, *6*, *10*

confRegCCR, 2, *4*, *8*, *10*
confRegUCRandUCCR, *3*, 3, *8*, *10*
constrOptim, *8*, *11*
crovData, 5

mdcp, *3*, *4*, 5, *9*, *10*
monoTestBonf, *3*, *4*, *8*, 8, *10*, *11*
monoTestConfReg, *3*, *4*, *8*, *9*, 9, *11*

plotCMLE, *3*, *4*, *8*–*10*, 10

vlgm, *3*, *4*, *9*, *10*