Package 'lcpm'

October 13, 2022

Type Package

Title Ordinal Outcomes: Generalized Linear Models with the Log Link

Version 0.1.1

Author Gurbakhshash Singh and Gordon Hilton Fick

Maintainer Gurbakhshash Singh <gsingh@ccsu.edu>

Description An implementation of the Log Cumulative Probability Model (LCPM) and Proportional Probability Model (PPM) for which the Maximum Likelihood Estimates are determined using constrained optimization.
This implementation accounts for the implicit constraints on the parameter space. Other features such as standard errors, z tests and p-values use standard methods adapted from the results based on constrained optimization.

License GPL-3

Encoding UTF-8

LazyData true

Depends plyr (>= 1.8.4), numDeriv (>= 2016.8-1), Matrix (>= 1.2-11), stats (>= 3.4.2)

RoxygenNote 7.0.2

NeedsCompilation no

Repository CRAN

Date/Publication 2020-01-09 07:10:02 UTC

R topics documented:

lcpm *Fitting a Log Cumulative Probability Model*

Description

lcpm provides the maximum likelihood estimate for ordinal outcomes (J>2 categories) and a Generalized Linear Model (GLM) with the log link without the assumption of proportionality. That is, lcpm determines the MLE for log[P(y <= j)]= cut_j + X beta_j subject to [cut_j-1 + X beta_j-1 <= cut_j + X beta_j] and [cut_j + X beta_j <=0]. This implementation uses constrOptim to determine the MLE and so the results account for the restricted parameter space.

Usage

```
lcpm(
  formula.linear,
  data,
  conf.level = 0.95,
  y.order = NULL,
  startval = NULL,
  less.than.0 = TRUE,
  control.list = NULL,
  eps.outer = NULL,
  ...
)
```

Arguments

formula.linear	an object of class "formula": a symbolic description of the linear model to be fitted.
data	dataframe containing the data in linear model.
conf.level	optional confidence level (1-alpha) defaulted to 0.95.
y.order	optional if y contains ordered integer categories 1:J. If y is not ordered integer 1:J then this is a vector with the ordinal values for y ranging from the lowest to largest ordinal outcome. See Examples below.
startval	optional vector of the starting values.
less.than.0	optional logical for constraint cut_j <= 0 for all j=1:(J-1). Default is TRUE.
control.list	optional list of controls for constrOptim
eps.outer	option for constrOptim
...	Additional arguments for built in functions

Value

list of class "lcpm" is returned containing:

coefficients	vector of the estimate of cut_j and beta_j

se	vector of the estimate of standard errors
vcov	matrix of the inverse of the negative Hessian
fitted.values	matrix of unique covariates and the corresponding estimate of the cumulative probabilities: exp(X %*% coefficients)
loglik	numerical value of the log-likelihood at the maximum likelihood estimate
barrier.value	value of mu in the log-barrier algorithm
outer.iterations	value of the number of outer iterations
formula	formula in the call of lcpm
startvalues	vector of the starting values for constrained optimization algorithm
proptest	Score test if a proportionality assumption is appropriate, includes test statistic (teststat), p-value (pval), df, and fitted proportional probability model (propmodel)

Note

A warning of MLE close to the boundary must be carefully considered. Data may have some structure that requires attention. Additionally, there is no imputation. Any NA results in complete row removal.

References

Singh, G; Fick, G.H. Ordinal outcomes: a cumulative probability model with the log link without an assumption of proportionality. Manuscript in preparation.

See Also

ppm

Examples

```
# Example below showing the use of y.order if outcome is not integers 1:J.
# See examples in ppm for an additional example


var_a <- c(rep(0,60),rep(1,60))
var_b <- c(rep(0,90),rep(1,30))
y1<-c(rep(2,5),rep(3,10),rep(5,5),rep(10,10),
rep(2,5),rep(3,10),rep(5,10),rep(10,5),
rep(2,10),rep(3,5),rep(5,5),rep(10,10),
rep(2,10),rep(3,5),rep(5,10),rep(10,5))

testdata<-data.frame(y=y1,var_a=var_a,var_b=var_b)

# LCPM estimates for non-proportional model
test1<-lcpm(y ~ var_a + var_b, data=testdata, y.order=c(2,3,5,10))
summary(test1)

# The proportional probability model used for the score test
summary(test1$proptest$propmodel)
```

lcpmMinusloglik — *The Negative Log-Likelihood Function for a Log Cumulative Probability Model*

Description

lcpmMinusloglik provides the negative of the log-likelihood function for a Generalized Linear Model with a log link and ordinal outcomes to be minimized in functions lcpm and ppm.

Usage

```
lcpmMinusloglik(betapar, Xa1, XaJ, Xaj1, Xaj2)
```

Arguments

betapar	a vector of values.
Xa1	matrix of covariates for all subjects with the lowest ordinal outcome value 1.
XaJ	matrix of covariates for all subjects with the largest ordinal outcome value J.
Xaj1	matrix of covariates for all subjects with the ordinal outcomes with value 1 < j < J.
Xaj2	matrix of covariates for all subjects with the ordinal outcome with value 1 < j < J but lagged by 1.

Value

value of the negative log-likelihood evaluated at betapar

ppm — *Fitting a Proportional Probability Model*

Description

ppm provides the maximum likelihood estimate for ordinal outcomes (J>2 categories) and a Generalized Linear Model with the log link with the assumption of proportionality. That is, ppm determines the MLE for log[P(y <= j)]= cut_j + X beta subject to [cut_j-1 <= cut_j] and [cut_j + X beta <=0]. This implementation uses constrOptim to determine the MLE and so the results should correctly account for the restricted parameter space. A proposed test for proportionality is included in lcpm.

Usage

```
ppm(
  formula.linear,
  data,
  conf.level = 0.95,
  y.order = NULL,
  startval = NULL,
  less.than.0 = TRUE,
  control.list = NULL,
  eps.outer = NULL,
  ...
)
```

Arguments

formula.linear	an object of class "formula": a symbolic description of the linear model to be fitted.
data	dataframe containing the data in linear model.
conf.level	optional confidence level (1-alpha) defaulted to 0.95.
y.order	optional if y contains ordered integer categories 1:J. If y is not ordered integer 1:J then this is a vector with the ordinal values for y ranging from the lowest to largest ordinal outcome. See Examples below.
startval	optional vector of the starting values.
less.than.0	optional logical for constraint cut_j <= 0 for all j=1:(J-1). Default is TRUE.
control.list	optional list of controls for constrOptim.
eps.outer	option for constrOptim.
...	Additional arguments for built in functions.

Value

list of class "ppm" is returned containing:

coefficients	vector of the estimate of cut_j and beta
se	vector of the estimate of standard errors
vcov	matrix of the inverse of the negative Hessian
fitted.values	matrix of unique covariates and the corresponding estimate of the cumulative probabilities: exp(X %*% coefficients)
loglik	numerical value of the log-likelihood at the maximum likelihood estimate
barrier.value	value of mu in the log-barrier algorithm
outer.iterations	value of the number of outer iterations
formula	formula in the call of ppm
startvalues	vector of the starting values for constrained optimization algorithm

Note

A warning of MLE close to the boundary must be carefully considered. Data may have some structure that requires attention. Additionally, there is no imputation. Any NA results in complete row removal.

References

Singh, G; Fick, G.H. (accepted) Ordinal outcomes: a cumulative probability model with the log link and an assumption of proportionality. Statistics in Medicine.

See Also

lcpm

Examples

```
# 2 examples below showing the use of y.order if outcome are not integers 1:J.

# Example 1:

var_a <- c(rep(0,60),rep(1,60))
var_b <- c(rep(0,90),rep(1,30))
y1<-c(rep(2,5),rep(3,10),rep(5,5),rep(10,10),
rep(2,5),rep(3,10),rep(5,10),rep(10,5),
rep(2,10),rep(3,5),rep(5,5),rep(10,10),
rep(2,10),rep(3,5),rep(5,10),rep(10,5))

testdata<-data.frame(y=y1,var_a=var_a,var_b=var_b)

# PPM estimates for proportional model
test1<-ppm( y ~ var_a + var_b, data=testdata, y.order=c(2,3,5,10))
summary(test1)

# Example 2:

y2<-c(rep("a",5),rep("b",10),rep("c",5),rep("d",10),
rep("a",5),rep("b",10),rep("c",10),rep("d",5),
rep("a",10),rep("b",5),rep("c",5),rep("d",10),
rep("a",10),rep("b",5),rep("c",10),rep("d",5))
testdata2<-data.frame(y=y2,var_a=var_a,var_b=var_b)
test2<-ppm(y~var_a + var_b , data=testdata2, y.order=c("a","b","c","d"))
summary(test2)
```

Index