

Package: clodr (via r-universe)

June 9, 2026

Type Package

Title Composite Likelihood Inference and Diagnostics for Replicated Spatial Ordinal Data

Version 1.7.2

Maintainer Ting Fung Ma <tingfung@mailbox.sc.edu>

Description Composite likelihood parameter estimate and asymptotic covariance matrix are calculated for the spatial ordinal data with replications, where spatial ordinal response with covariate and both spatial exponential covariance within subject and independent and identically distributed measurement error. Parameter estimation can be performed by either solving the gradient function or maximizing composite log-likelihood. Parametric bootstrapping is used to estimate the Godambe information matrix and hence the asymptotic standard error and covariance matrix with parallel processing option. Moreover, the proposed surrogate residual, which extends the results of Liu and Zhang (2017) <doi:10.1080/01621459.2017.1292915>, can act as a useful tool for model diagnostics.

License GPL-2

Imports pbivnorm (>= 0.6.0), MASS (>= 7.3-45), rootSolve (>= 1.7), parallel, doParallel (>= 1.0.11), foreach (>= 1.2.0), tmvmixnorm (>= 1.0.2), utils, stats

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Ting Fung Ma [cre, aut], Pingping Wang [aut], Jun Zhu [aut], Dipankar Bandyopadhyay [ctb], Yincai Tang [ctb]

Repository <https://ralphma1203.r-universe.dev>

Date/Publication 2026-05-09 18:09:55 UTC

RemoteUrl <https://github.com/cran/clodr>

RemoteRef HEAD

RemoteSha b91bfad2bcae0d392f8573f640c9e15d624d4b31

Contents

cl	2
cl.rord	3
cl_l	4
cle.rord	5
clc	7
merge.list	8
sim.rord	8
surrogate.residual	10

Index	13
--------------	-----------

cl	<i>Composite Likelihood Calculation for Spatial Ordinal Data without Replications (for implementation)</i>
----	--

Description

cl Calculate the negative composite log-likelihood value and score function for a particular subject given parameter value and other input variables.

Usage

```
cl(theta, y, X, dwdv, cmwdv, lt, wn, base, J, p)
```

Arguments

theta	a vector of parameter value.
y	a vector of observation for the subject.
X	covariate for the particular subject.
dwdv	corresponding distance of selected pair.
cmwdv	combination of the pairs included into the composite likelihood.
lt	number of parameter (i.e. length of theta).
wn	number of pairs with distance.
base	identity matrix with dimension J+1.
J	number of category among (ALL) observed response.
p	number of covariate (i.e. number of column of X).

Value

cl returns a list: composite log-likelihood value and a vector of first-order partial derivatives for theta.

cl.rord	<i>Composite Likelihood Calculation for Replciations of Spatial Ordinal Data (for illustration)</i>
---------	---

Description

cl.rord Calculate the negative composite log-likelihood value for replications of spatial ordinal data at given value of parameter value. Note that this function is not directly used in cle.rord but illustration only.

Usage

```
cl.rord(theta, response, covar, location, radius = 4)
```

Arguments

theta	a vector of parameter value
response	a matrix of observation (row: spatial site and column: subject).
covar	regression (design) matrix, including intercepts.
location	a matrix contains spatial location of sites within each subject
radius	radius for selecting pairs for the composite likelihood estimation.

Value

cl.rord returns a list: negative composite log-likelihood, a vector of first-order partial derivatives for theta.

Examples

```
set.seed(1203)
n.subject <- 10
n.lat <- n.lon <- 10
n.site <- n.lat*n.lon

beta <- c(1,2,-1) # First 1 here is the intercept
midalpha <- c(1.15, 2.18) ; sigma2 <- 0.7 ; phi <- 0.8

true = c(midalpha,beta,sigma2,phi)

Xi = rnorm(n.subject,0,1) ; Xj <- rbinom(n.site,1,0.6)

VV <- matrix(NA, nrow = n.subject*n.site, ncol = 3)

for(i in 1:n.subject){ for(j in 1:n.site){
  WV[(i-1)*n.site+j,] <- c(1,Xi[i],Xj[j])
}
}
```

```
location = cbind(rep(seq(1,n.lat,length=n.lat),n.lat),rep(1:n.lon, each=n.lon))
sim.data <- sim.rord(n.subject, n.site, n.rep = 2, midalpha, beta, sigma2, phi, covar=VV, location)

cl.rord(theta=true,response=sim.data[[1]], covar=VV, location, radius = 4)
```

cl_1	<i>Composite Likelihood Calculation for Spatial Ordinal Data without Replications (for implmentation)</i>
------	---

Description

cl_1 Calculate the negative composite log-likelihood value for a particular subject given parameter value and other input variables

Usage

```
cl_1(theta, y, X, dwdv, cmwdv, lt, wn, base, J, p)
```

Arguments

theta	a vector of parameter value
y	a vector of observation for the subject.
X	covariate for the particular subject
dwdv	corresponding distance of selected pair
cmwdv	combination of the pairs included into the composite likelihood
lt	number of parameter (i.e. length of theta)
wn	number of pairs
base	identity matrix with dimension J+1
J	number of category among (ALL) observed response.
p	number of covariate (i.e. number of column of X)

Value

cl returns a list: composite log-likelihood value and a vector of first-order partial derivatives for theta.

cle.rord	<i>Composite Likelihood Estimation for Replciations of Spatial Ordinal Data</i>
----------	---

Description

cle.rord Estimate parameters (including regression coefficient and cutoff) for replications of spatial ordinal data using pairwise likelihood approach.

Usage

```
cle.rord(
  response,
  covar,
  location,
  radius = 4,
  n.sim = 100,
  output = TRUE,
  SE = TRUE,
  parallel = FALSE,
  n.core = max(detectCores()/2, 1),
  ini.sp = c(0.5, 0.5),
  est.method = TRUE,
  maxiter = 100,
  rtol = 1e-06,
  factr = 1e+07
)
```

Arguments

response	a matrix of observation (row: spatial site and column: subject).
covar	regression (design) matrix, including intercepts.
location	a matrix contains spatial location of sites within each subject.
radius	radius for selecting pairs for the composite likelihood estimation.
n.sim	number of simulation used for parametric bootstrapping (and hence used for asymptotic variance and standard error).
output	logical flag indicates whether printing out result (default: TRUE).
SE	logical flag for detailed output.
parallel	logical flag indicates using parallel processing (default: FALSE).
n.core	number of physical cores used for parallel processing (when parallel is TRUE, default value is $\max(\text{detectCores}()/2, 1)$).
ini.sp	initial estimate for spatial parameter, ϕ, σ^2 (default: $\text{c}(0.5, 0.5)$).
est.method	logical flag (default) TRUE for rootsolve and FALSE for L-BFGS-B.

maxiter	maximum number of iterations in the root solving of gradient function (default: 100).
rto1	relative error tolerance in the root solving of gradient function (default: 1e-6).
factr	reduction in the objective (-logCL) within this factor of the machine tolerance for L-BFGS-B (default: 1e7).

Details

Given vector of ordinal responses, the design matrix, spatial location for sites, weight radius (for pair selection), and the prespecified number of simulation used for estimating the Godambe information matrix. Initial estimate is obtained by fitting model without spatial dependence (using `MASS::polr()`) and optional guess of spatial parameters. The function first estimates parameters of interest by either solving the gradient of composite log-likelihood using `rootSolve::multiroot()` or maximize the composite log-likelihood by `optim(..., method="L-BFGS-B")`. The asymptotic covariance matrix and standard error of parameters are then estimated by parametric bootstrapping. Although the default root solving option is typically more efficient, it may encounter runtime error if negative value of ϕ is evaluated (and L-BFGS-B approach should be used).

Value

`cle.rord` returns a list contains:

`vec.par`: a vector of estimator for $\theta = (\alpha, \beta, \phi, \sigma^2)$;

`vec.se`: a vector of standard error for the estimator;

`mat.asyvar`: estimated asymptotic covariance matrix $H^{-1}(\theta)J(\theta)H^{-1}(\theta)$ for the estimator;

`mat.Hessian`: Hessian matrix at the parameter estimate;

`mat.J`: Sensitivity matrix estimated by parametric bootstrapping; and

`CLIC`: Composite likelihood information criterion (see help manual of `clic()` for detail).

Examples

```
set.seed(1228)
n.subject <- 20
n.lat <- n.lon <- 10
n.site <- n.lat*n.lon

beta <- c(1,2,-1) # First 1 here is the intercept
midalpha <- c(1.15, 2.18) ; phi <- 0.6 ; sigma2 <- 0.7

true <- c(midalpha,beta,phi,sigma2)

Xi <- rnorm(n.subject,0,1) ; Xj <- rbinom(n.site,1,0.6)

VV <- matrix(NA, nrow = n.subject*n.site, ncol = 3)

for(i in 1:n.subject){ for(j in 1:n.site){
  VV[(i-1)*n.site+j,] <- c(1,Xi[i],Xj[j])
}
}
```

```

location <- cbind(rep(seq(1,n.lat,length=n.lat),n.lat),rep(1:n.lon, each=n.lon))
sim.data <- sim.rord(n.subject, n.site, n.rep = 2, midalpha, beta, phi, sigma2, covar=VV, location)

options(digits=3)
result <- cle.rord(response=sim.data[[1]], covar=VV,
  location = location ,radius = 4, n.sim = 100, output = TRUE, parallel=TRUE, n.core =2)
result$vec.par
# alpha2 alpha3 beta0 beta1 beta2 phi sigma^2
# 1.249 2.319 1.169 1.990 -1.000 0.668 0.678

result$vec.se
# alpha2 alpha3 beta0 beta1 beta2 phi sigma^2
# 0.0704 0.1201 0.1370 0.2272 0.0767 0.0346 0.1050

```

clic

Composite likelihood Information Criterion

Description

clic Calculating the Composite likelihood information criterion proposed by Varin and Vidoni (2005)

Usage

```
clic(logCL, mat.hessian, mat.J)
```

Arguments

logCL	value of composite log-likelihood.
mat.hessian	hessian matrix.
mat.J	Sensitivity matrix

Details

Varin and Vidoni (2005) proposed the information criterion in the form: $-2 * \log CL(\theta) + 2 * \text{trace}(H^{-1}(\theta)J(\theta))$

Value

CLIC: Composite likelihood information criterion proposed by Varin and Vidoni (2005)

clic: Composite likelihood information criterion proposed by Varin and Vidoni (2005)

References

Varin, C. and Vidoni, P. (2005) A note on composite likelihood inference and model selection. *Biometrika* 92: 519–528.

merge.list	<i>Composite Likelihood Calculation for Spatial Ordinal Data without Replications (for implementation)</i>
------------	--

Description

Composite Likelihood Calculation for Spatial Ordinal Data without Replications (for implementation)

Usage

```
## S3 method for class 'list'
merge(x, y = NULL, mergeUnnamed = TRUE)
```

Arguments

x	an object to be merged into list of object.
y	an object to be merged into list.
mergeUnnamed	select an element if it has a.) an empty name in y and mergeUnnamed is true or b.) a name <code>_not_</code> contained in x

Value

merge.list returns a list: a merged list

sim.rord	<i>Simulation of Replciations of Spatial Ordinal Data</i>
----------	---

Description

sim.rord Simulate replications of spatial ordinal data

Usage

```
sim.rord(
  n.subject,
  n.site,
  n.rep = 100,
  midalpha,
  beta,
  phi,
  sigma2,
  covar,
  location
)
```

Arguments

n.subject	number of subjects.
n.site	number of spatial sites for each subject.
n.rep	number of simulation. Parameter inputs include:
midalpha	cutoff parameter (excluding -Inf and +Inf);
beta	regression coefficient;
phi	dependence parameter for spatial dependence; and
sigma2	sigma ² (variance) for the spatial dependence.
covar	regression (design) matrix, including intercepts.
location	a matrix contains spatial location of sites within each subject.

Value

sim.rord returns a list (length n.rep) of matrix (n.subject*n.site) with the underlying parameter as inputs.

Examples

```
set.seed(1203)
n.subject <- 100
n.lat <- n.lon <- 10
n.site <- n.lat*n.lon

beta <- c(1,2,-1) # First 1 here is the intercept
midalpha <- c(1.15, 2.18) ; phi <- 0.8 ; sigma2 <- 0.7

true <- c(midalpha,beta,sigma2,phi)

Xi <- rnorm(n.subject,0,1) ; Xj <- rbinom(n.site,1,0.6)

VV <- matrix(NA, nrow = n.subject*n.site, ncol = 3)

for(i in 1:n.subject){ for(j in 1:n.site){
```

```

      WV[(i-1)*n.site+j,] <- c(1,Xi[i],Xj[j])
    }
  }

location <- cbind(rep(seq(1,n.lat,length=n.lat),n.lat),rep(1:n.lon, each=n.lon))
sim.data <- sim.rord(n.subject, n.site, n.rep = 2, midalpha, beta, phi, sigma2, covar=VV, location)

length(sim.data)
head(sim.data[[1]])
dim(sim.data[[1]])
hist(sim.data[[1]])

```

surrogate.residual *Surrogate Residuals for Replciations of Spatial Ordinal Data*

Description

surrogate.residual simulate the surrogate residual with the the given parameter value and co-variate for model diagnostics.

Usage

```

surrogate.residual(
  response,
  covar,
  location,
  seed = NULL,
  midalpha,
  beta,
  sigma2,
  phi,
  burn.in = 20,
  output = TRUE
)

```

Arguments

response	a matrix of observation (row: spatial site and column: subject).
covar	regression (design) matrix, including intercepts.
location	a matrix contains spatial location of sites within each subject.
seed	seed input for simulation (default =NULL). Parameter values:
midalpha	cutoff for latent ordinal response.
beta	regression coefficient for covar.
sigma2	σ^2 for exponential covariance.
phi	spatial correlation for exponential covariance.
burn.in	burn-in length (i.e. declaring the initial sample).
output	logical flag indicates whether printing out result (default: TRUE).

Details

Given vector of observed responses, the design matrix, spatial location for sites and parameter value, raw surrogate residuals are simulated using an efficient Gibbs sampling, which can be used for model diagnostics. When the fitted model is correct, the raw surrogate residuals among subjects should follow multivariate normal with mean 0 and covariance Sigma. If the model is correct, residual plot should be close to a null plot or random scatter. For example, it can be used to check the potential missing in covariate, non-linearity of covariate and outliers. In particular for the example below, the residual plot shows that linearity of X_i is adequate for the model.

Value

surrogate.residual returns a (no. spatial site * no. subject) matrix contains raw surrogate residuals with element corresponds to the response matrix.

Examples

```
set.seed(1228)
n.subject <- 50
n.lat <- n.lon <- 10
n.site <- n.lat*n.lon

beta <- c(1,2,-1) # First 1 here is the intercept
midalpha <- c(1.15, 2.18) ; phi <- 0.6 ; sigma2 <- 0.7

true <- c(midalpha,beta,phi,sigma2)

Xi <- rnorm(n.subject,0,1) ; Xj <- rbinom(n.site,1,0.6)

VV <- matrix(NA, nrow = n.subject*n.site, ncol = 3)

for(i in 1:n.subject){ for(j in 1:n.site){
  VV[(i-1)*n.site+j,] <- c(1,Xi[i],Xj[j])
}
}

location <- cbind(rep(seq(1,n.lat,length=n.lat),n.lat),rep(1:n.lon, each=n.lon))
response <- sim.rord(n.subject, n.site, n.rep = 1,
midalpha, beta, phi, sigma2, covar=VV, location)[[1]]

# Example for linearity of covariate
sur.resid <- surrogate.residual(response, covar=VV, location, seed =1,
midalpha, beta, sigma2, phi,
burn.in=20, output = TRUE)

scatter.smooth(rep(Xi,each=n.site),c(sur.resid),
main="Surrogate residual against Xi", xlab="Xi", ylab="Surrogate residual",
lpars = list(col = "red", lwd = 3, lty = 2))

abline(h=0, col="blue")
```


Index

`cl`, 2
`cl.rord`, 3
`cl_l`, 4
`cle.rord`, 5
`cllc`, 7

`merge.list`, 8

`sim.rord`, 8
`surrogate.residual`, 10