Package: pmlsp (via r-universe)

November 13, 2026

Title Partial Maximum Likelihood Estimation of Spatial Probit Models

Version 1.0.1

Description Estimate spatial autoregressive nonlinear probit models with and without autoregressive disturbances using partial maximum likelihood estimation. Estimation and inference regarding marginal effects is also possible. For more details see Bille and Leorato (2020) <doi:10.1080/07474938.2019.1682314>.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports abind, Matrix, matrixcalc, maxLik, methods, minqa, mvtnorm, numDeriv, qrng, spatialreg, spdep, stats, utils

Depends R (>= 3.5)

LazyData true

URL https://github.com/d-spinelli/pmlsp

BugReports https://github.com/d-spinelli/pmlsp/issues

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://d-spinelli.r-universe.dev

Date/Publication 2026-10-14 08:39:22 UTC

RemoteUrl https://github.com/d-spinelli/pmlsp

RemoteRef HEAD

RemoteSha 98a4f476cafd656240b9dd47fadac82e0329c06c

Contents

ape 2
eurostat 3
pmlsbp 4
predict.pmlsprobit 7

Index **9**

---

ape *Average partial (i.e., marginal) total, direct and indirect effects from a pmlsprobit object*

Description

Average partial (i.e., marginal) total, direct and indirect effects from a pmlsprobit object

Usage

```
ape(mod, variables = NULL)
```

Arguments

mod	Object of class inheriting from "pmlsbp"
variables	a string including the variable names of the covariates for which the marginal effects should be calculated using numeric derivatives. It is valid only if type="me". This option should be used only if the model contains non-linear terms in the specification $X\beta$

Value

A numeric vector with predicted values.

Examples

```
# load the data
data(oldcol, package="spdep")

# create binary response variable
COL.OLD$y<-as.numeric(COL.OLD$CRIME>35)

# prepare the spatial weight matrix
listw <- spdep::nb2listw(COL.nb, style="W")

set.seed(857489)

# fit a SAR model
mod<-pmlsbp (y~HOVAL+INC, COL.OLD, W=listw, model = 'SAR',print.level = 2,
         mvtnorm_control=list(M=25e2, sim_type="mc" , tol = .Machine$double.eps, fast = FALSE),
         grouping = 7, na.action=na.omit, spectral = TRUE, iterlim=1e5, tol.solve = 1e-300)
```

```
# estimate the average partial effects
ape(mod)
```

`eurostat` *Eurostat employment data*

Description

The `eurostat` data frame has 312 row and 10 variables. Unit of analysis: 312 regions in Europe. Additionally, the `eurostat.nb` includes the neighbour list based on a consider a k-nearest neighbour approach with k = 11

Usage

```
data(eurostat)
```

Format

A data frame with 312 rows and 10 variables:

`nuts_id` Region identifier

`country_id` Country identfier

`employment` Employement rate in percentage

`gdp` GDP per capita

`isced_02` percentage of residents aged 25-64 with secondary school achievement or lower

`isced_34` percentage of residents aged 25-64 with high school achievement or lower

`isced_58` percentage of residents aged 25-64 with bachelor degree or higher

`long` Longitude

`lat` Latitude

Source

https://ec.europa.eu/eurostat/data/database

pmlsbp *pmlsbp: Fitting Spatial Probit Models*

Description

Estimation of Partial Maximum Likelihood Spatial Autoregressive (SAR) Probit Models and Spatial Autoregressive Probit Models with Autoregressive Disturbances (SARAR). The model structure for the SARAR model is:

$$y^* = \rho W y + X\beta + W_2 Z\gamma + \mathrm{u}$$
$$u = \lambda M u + \epsilon$$

where $\epsilon \sim N(0, \sigma^2)$ and $y = 1(y^* > 0)$. The SAR model assumes $\lambda = 0$

Usage

```
pmlsbp(formula,data, model="SAR", grouping=2, W=NULL,
zero.policy =spatialreg::get.ZeroPolicyOption(),
M=NULL, formula_xlag=NULL, W2=NULL,  method_inv="solve",
start=NULL, subset=NULL, na.action=na.fail,qu=Inf, iterlim=1000,
mvtnorm_control=list(M=1e4, sim_type="qmc" , tol = .Machine$double.eps, fast = FALSE),
finalHessian=ifelse( method=="bhhh",method,TRUE), method="bhhh",print.level=2,
vce.type='asy' ,  Conley=list(coords=NULL,LM=2),nBoot=1e3 , spectral=F,
tol.solve= .Machine$double.eps, version=0, ...)

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

## S3 method for class 'pmlsprobit'
vcov(object, ...)

## S3 method for class 'pmlsprobit'
coef(object, ...)

## S3 method for class 'pmlsprobit'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

formula	an object of class "formula": a symbolic description of the model to be fit.
data	an optional data frame containing the variables in the model. By default the variables are taken from the environment which the function is called.
model	a character ("SAR" or "SARAR") defining the model to be estimated. The default is "SAR".

grouping	an integer defining the number of observations to include in the tuples to be used for estimation, the default is 2.
W	a listw object created for example by nb2listw
zero.policy	default NULL, use global option value; if TRUE assign zero to the lagged value of zones without neighbours, if FALSE (default) assign NA - causing pmlsbp to terminate with an error (see also set.ZeroPolicyOption) and the same option in GMerrorsar)
M	a listw object created for example by nb2listw to be used for the model with spatial autoregressive disturbances. This is relevant only if model="SARAR".
formula_xlag	an object of class "formula": a symbolic description of the covariates to be spatially lagged.
W2	a listw object created for example by nb2listw to be used to create the spatial weight matrix to multiply the covariates. This is relevant only if formula_xlag is not NULL.
method_inv	a character for setting the inversion of $I - \rho * W$, either "solve", "chol" or "fast". "solve" uses solve, "chol" performs the inversion of the matrix from its Choleski decomposition (see chol2inv), "fast" uses an approximation that requires option qu
start	numeric vector, initial value of parameters.
subset	an optional vector specifying a subset of observations to be used for fitting
na.action	a function (default na.fail), can also be na.omit or na.exclude with consequences for residuals and fitted values - in these cases the weights list will be subset to remove NAs in the data. It may be necessary to set zero.policy to TRUE because this subsetting may create no-neighbour observations. Note that only weights lists created without using the glist argument to nb2listw may be subsetted.
qu	an integer to be used only if method_inv="fast"
iterlim	an intenger. Maximum number of iterations to be used by maxLik
mvtnorm_control	an optional object of class "list" that includes the controls for the calculation of the multivariate normal probabilities. The list members should include M, rmtol and fast (see lpmvnorm) and simtype, either "mc" for Monte-Carlo procedure or "qmc", quasi-Monte-Carlo procedure) The default is list(M=25e3, sim_type="mc" , tol = .Machine$double.eps, fast = FALSE)
finalHessian	how (and if) to calculate the final Hessian. Either FALSE (not calculate), TRUE (use analytic/numeric Hessian) or "bhhh"/"BHHH" for information equality approach (see maxLik).
method	maximisation method to be used by maxLik. The default is method="bhhh" Broyden-Fletcher-Goldfarb-Shanno as implemented in R. See also the same option in (see maxLik).
print.level	Integer, debugging information. Larger number prints more details (see maxLik).
vce.type	an optional character specifying the type of variance covariance estimation (VCE) to be performed. It can be either "asy" for the asymptotic VCE, "bootstrap" for bootstrap VCE or "mConley" for the modified Conley approach. The default is "asy".

Conley	a list with two members, which is relevant only for vce.type="mConley". The first member (coords) is a matrix with the coordinates of each observation; the second member (LM) is the number of neighbors to be used. The default is list(coords=NULL,LM=2)
nBoot	an integer, the number of bootstrap samples to be used for vce.type="bootstrap". It is relevant only if vce.type="bootstrap".
spectral	a boolean. If spectral=TRUE uses the standardization of the W matrix based on the spectral decomposition
tol.solve	the tolerance to be used by solve when inverting matrices. The default is .Machine$double.eps
version	an integer that affects the calculation only if the number of observations is even and grouping=2. If version=0 the bivariate normal integrals are calculated analytically instead of being calculated using mvtnorm.The default is method="bhhh".
...	additional arguments to be passed to maxLik
object	a "pmlsprobit" object
x	a "pmlsprobit" object
digits	integer, used for number formatting the decimal digits

Value

pmlsbp returns an object of class "pmlsprobit". The function summary is used to obtain and print a summary of the results. A object of class "pmlsprobit" is list containing the following components:

beta	a named vector including the β and γ, the coefficents attached to the variables resulting from formula and for
call	the call
code	
estimate	a named vector of model estimates
f	a list including useful matrices to be used in predict
maximum	the value of the partial log-likelihood given the data and the model estimates
message	convergence message
model	
model.type	either "SAR" or "SARAR"
rho	a named scalar for the ρ parameter
slx	TRUE if the model includes spatially lagged covariates
start	a named vector, the starting values used in estimation
terms	the terms object used.
vcov	the variance-covariance matrix of the estimates
W	matrix W
lambda	a named scalar for parameter $lambda$ (only for SARAR models)
M	matrix M (only for SARAR models)
W2	matrix W_2 (only if slx=TRUE)

Functions

- `summary(pmlsprobit)`: computes and returns a list of summary statistics of the fitted spatial probit model
- `vcov(pmlsprobit)`: returns the variance covariance matrix of a `"pmlsprobit"` object
- `coef(pmlsprobit)`: returns the coefficients estimate from a `"pmlsprobit"` object
- `print(pmlsprobit)`: prints the call and the coefficient estimates of a `"pmlsprobit"` object

References

Bille', A. G., & Leorato, S. (2020). Partial ML estimation for spatial autoregressive nonlinear probit models with autoregressive disturbances. Econometric Reviews, 39(5), 437-475.

Examples

```
# load the data
data(oldcol, package="spdep")

# create binary response variable
COL.OLD$y<-as.numeric(COL.OLD$CRIME>35)

# prepare the spatial weight matrix
listw <- spdep::nb2listw(COL.nb, style="W")

set.seed(857489)

# fit a SAR model
mod<-pmlsbp (y~HOVAL+INC, COL.OLD, W=listw, model = 'SAR',print.level = 2,
              grouping = 7, na.action=na.omit, spectral = TRUE, iterlim=1e5)
```

---

predict.pmlsprobit *Predict method for pmlsprobit objects*

Description

Predicted values based on pmlsprobit object

Usage

```
## S3 method for class 'pmlsprobit'
predict(
  object,
  type = "prob",
  newdata = NULL,
  variables = NULL,
  delta.method = F,
  ...
)
```

Arguments

object	a "pmlsprobit" object
type	"prob" for the probability $P(Y_i = 1)$ or "me" for the direct, indirect and total average marginal effect for each observation
newdata	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
variables	a string including the variable names of the covariates for which the marginal effects should be calculated using numeric derivatives. It is valid only if type="me". This option should be used only if the model contains non-linear terms in the specification $X\beta$.
delta.method	a boolean. If TRUE the standard errors of the predictions are calculated. It is valid only if type="me"
...	additional arguments

Value

A numeric vector with predicted values.

Examples

```
# load the data
data(oldcol, package="spdep")

# create binary response variable
COL.OLD$y<-as.numeric(COL.OLD$CRIME>35)

# prepare the spatial weight matrix
listw <- spdep::nb2listw(COL.nb, style="W")

set.seed(857489)

# fit a SAR model
mod<-pmlsbp (y~HOVAL+INC, COL.OLD, W=listw, model = 'SAR',print.level = 2,
         mvtnorm_control=list(M=25e2, sim_type="mc" , tol = .Machine$double.eps, fast = FALSE),
         grouping = 7, na.action=na.omit, spectral = TRUE, iterlim=1e5, tol.solve = 1e-300)

# estimate the average partial effects
prob <- predict(mod)
head(prob)
```

Index

∗ **datasets**
eurostat, 3

ape, 2

chol2inv, *5*
coef.pmlsprobit *(*pmlsbp*)*, 4

eurostat, 3

GMerrorsar, *5*

lpmvnorm, *5*

maxLik, *5*

pmlsbp, 4
predict.pmlsprobit, 7
print.pmlsprobit *(*pmlsbp*)*, 4

set.ZeroPolicyOption, *5*
summary.pmlsprobit *(*pmlsbp*)*, 4

vcov.pmlsprobit *(*pmlsbp*)*, 4