Package 'ProbitSpatial'

January 20, 2025

Type Package

Title Probit with Spatial Dependence, SAR, SEM and SARAR Models

Version 1.1

Date 2021-10-06

Maintainer Davide Martinetti <davide.martinetti@inrae.fr>

Description Fast estimation of binomial spatial probit regression models with spatial autocorrelation for big datasets.

License GPL (>= 2)

Imports numDeriv, RANN, Rcpp (>= 1.0.4.6), methods

Depends Matrix

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.1.1

NeedsCompilation yes

Author Davide Martinetti [aut, cre] (<https://orcid.org/0000-0003-2047-1793>),
Ghislain Geniaux [aut]

Repository CRAN

Date/Publication 2021-06-30 08:50:02 UTC

Contents

`ProbitSpatial-package` *Probit with Spatial Dependence, SAR, SEM, and SARAR Models.*

Description

`ProbitSpatial` package allows to fit spatial autoregressive (SAR) and spatial error (SEM) probit models. It also provides functions to simulated spatial binary data, an empirical data set and different methods for the diagnostic of the estimated model.

Details

The main function of this package is `ProbitSpatialFit`. It allows to fit both SAR and SEM models for big datasets in a reasonable time. The function is based on the maximisation of the approximate likelihood function. The approximation is inspired by the Mendell and Elston algorithm for computing multivariate normal probabilities and take advantage of the sparsity of the spatial weight matrix. Two methods are available for the estimation of the model parameter: the first one is known as conditional method (see Case (1992)) and performs relatively well in terms of accuracy of the estimated parameters and is very rapid. The second method, that minimises the full-log-likelihood, is slower but it should be more accurate. Monte Carlo experiments on simulated data reported in Martinetti and Geniaux (2017) showed that the full-log-likelihood approach is not always overperforming the conditional method in terms of accuracy. At the present stage, our suggestion is to use the conditional method for a first estimation and only attempt the full-likelihood approach in a second moment, when the dataset size is not bigger than a few thousands.

Another feature of the `ProbitSpatialFit` function is the possibility to fit the model using the precision matrix instead of the variance-covariance matrix, since it is usually sparser and hence allows faster computations (see LeSage and Pace (2009)).

The output of `ProbitSpatialFit` function is an object of class `ProbitSpatial`, for which the methods `residuals`, `fitted`, `effects`, `predict` and `coef` are available.

The package also contains the function `sim_binomial_probit` that allows to simulate data samples of both SAR and SEM models. It can be used to replicate the Monte Carlo experiments reported in Martinetti and Geniaux (2017) as well as the experiment of Calabrese and Elkink (2014). An empirical data set `Katrina` on the reopening decisions of firms in the aftermath of the Katrina Hurricane in New Orleans is also available (LeSage et al.(2011)).

Other packages in CRAN repository on the same subject are `McSpatial` (McMillen (2013)) and `spatialprobit` (Wilhelm and Godinho de Matos (2013)).

The core functions of the present package have been coded using the `Rcpp` and `RcppEigen` libraries (Bates and Eddelbuettel (2013)), that allow direct interchange of rich R objects between R and C++.

Author(s)

Davide Martinetti `<davide.martinetti@inra.fr>` and Ghislain Geniaux `<ghislain.geniaux@inra.fr>`

References

Bates and Eddelbuettel (2013) D. Bates and D. Eddelbuettel. Fast and elegant numerical linear algebra using the RcppEigen package. *Journal of Statistical Software* 52, 1–24, 2013.

Case (1992) A. C. Case. Neighborhood Influence and Technological Change. *Regional Science and Urban Economics* 22, 491–508, 1992.

Calabrese and Elkink (2014) R. Calabrese and J.A. Elkink. Estimators of binary spatial autoregressive models: a Monte Carlo study. *Journal of Regional Science* 54, 664–687, 2014.

LeSage and Pace (2009) J. LeSage and R.K. Pace. *Introduction to Spatial Econometrics*, CRC Press, chapter 10.1.6, 2009.

LeSage et al. (2011) P. LeSage, R. K. Pace, N. Lam, R. Campanella and X. Liu. New Orleans business recovery in the aftermath of Hurricane Katrina. *Journal of the Royal Statistical Society A* 174, 1007–1027, 2011.

Martinetti and Geniaux (2017) D. Martinetti and G. Geniaux. Approximate likelihood estimation of spatial probit models. *Regional Science and Urban Economics* 64, 30-45, 2017.

McMillen (2013) D. McMillen. McSpatial: Nonparametric spatial data analysis. R package version 2.0, 2013.

Mendell and Elston (1974) N. Mendell and R. Elston. Multifactorial qualitative traits: genetic analysis and prediction of recurrence risks. *Biometrics* 30, 41–57, 1974.

Wilhelm and Godinho de Matos (2013) S. Wilhelm and M. Godinho de Matos. Estimating Spatial Probit Models in R. *The R Journal* 5, 130–143, 2013.

`coef.ProbitSpatial` *Estimated coefficients of a spatial probit model.*

Description

Returns the coefficients estimated by a `ProbitSpatial` model.

Usage

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

Arguments

`object`	an object of class `ProbitSpatial`.
`...`	ignored

Value

It returns the value of the estimated parameters.

`conditional_SARAR_UC` *Conditional SARAR UC.*

Description

Performs conditional estimation of SARAR model with variance-covariance matrix.

Usage

```
conditional_SARAR_UC(myenv)
```

Arguments

`myenv`	an `environment`.

Details

We discourage the direct use of this function.

Value

the log-likelihood and the estimated parameters.

`conditional_SARAR_UP` *Conditional SARAR UP.*

Description

Performs conditional estimation of SARAR model with precision matrix.

Usage

```
conditional_SARAR_UP(myenv)
```

Arguments

`myenv`	an `environment`.

Details

We discourage the direct use of this function.

Value

the log-likelihood and the estimated parameters.

conditional_SAR_UC *Conditional SAR UC.*

Description

Performs conditional estimation of SAR model with variance-covariance matrix.

Usage

```
conditional_SAR_UC(myenv)
```

Arguments

myenv	an environment.

Details

We discourage the direct use of this function.

Value

the log-likelihood and the estimated parameters.

conditional_SAR_UP *Conditional SAR UP.*

Description

Performs conditional estimation of SAR model with precision matrix.

Usage

```
conditional_SAR_UP(myenv)
```

Arguments

myenv	an environment.

Details

We discourage the direct use of this function.

Value

the log-likelihood and the estimated parameters.

conditional_SEM_UC *Conditional SEM UC.*

Description

Performs conditional estimation of SEM model with variance-covariance matrix.

Usage

```
conditional_SEM_UC(myenv)
```

Arguments

myenv	an environment.

Details

We discourage the direct use of this function.

Value

the log-likelihood and the estimated parameters.

conditional_SEM_UP *Conditional SEM UP.*

Description

Performs conditional estimation of SEM model with precision matrix.

Usage

```
conditional_SEM_UP(myenv)
```

Arguments

myenv	an environment.

Details

We discourage the direct use of this function.

Value

the log-likelihood and the estimated parameters.

effects_ProbitSpatial *Effects of a spatial probit model.*

Description

Returns the marginal effects of a `ProbitSpatial` model.

Usage

```
effects_ProbitSpatial(object)
```

Arguments

`object`	an object of class `ProbitSpatial`.

Details

The `effects` function has different outputs according to the DGP of the `ProbitSpatial` model:

`"SAR"` The marginal effects of a spatial autoregressive model are more complicated than usual measurements of impacts for non spatial models. Here we follow LeSage and Pace and propose the following summaries for impact measures:

- **Average direct effects:** the average over all the observations of the effects of the change of an explanatory variable of a single observation on the choice probability of that same observation.
- **Average indirect effects:** the average over all the observations of the effect of a change on a explanatory variable on the choice probability of the neighbouring observations.
- **Average total effects:** the sum of direct and indirect impacts.

`"SEM"` marginal effects should be interpreted as if it were a standard probit model.

Value

It returns the marginal effects of the estimated `ProbitSpatial` model.

References

J. LeSage and R.K. Pace. *Introduction to Spatial Econometrics*, CRC Press, chapter 10.1.6, 2009.

fitted.ProbitSpatial *Extract spatial probit model fitted values.*

Description

Extract the fitted values of a `ProbitSpatial` model.

Usage

```
## S3 method for class 'ProbitSpatial'
fitted(object, type = c("link", "response", "binary"), cut = 0.5, ...)
```

Arguments

`object`	an object of class `ProbitSpatial`.
`type`	the type of output: `"link"` the value of the latent variable. Default. `"response"` probability. `"binary"` binary 0/1 output.
`cut`	the threshold probability for the `"binary"` type. Default is 0.5.
`...`	ignored

Value

Returns the vector of fitted values of the `ProbitSpatial` model

generate_W *Generate a random spatial weight matrix.*

Description

Generate a spatial weight matrix of given size and number of nearest neighbors from randomly-located observations on the unit square.

Usage

```
generate_W(n, nneigh, seed=123)
```

Arguments

`n`	the size of the matrix.
`nneigh`	the number of nearest neighbors.
`seed`	an integer to set the seed for the random generated locations.

Details

The output matrix has zero diagonal and it is row-standardised. The `n` observations are allocated randomly in the unit square. For each observation, the `nneigh` closests observations w.r.t. the Euclidean distance are assigned with a weight equal to 1/`nneigh`.

Value

a matrix of class `dgCMatrix` (sparse matrix).

See Also

`sim_binomial_probit`.

Examples

```
W <- generate_W(100,4,seed=12)
```

Katrina *New Orleans business recovery in the aftermath of Hurricane Katrina.*

Description

This dataset has been used in the LeSage et al. (2011) paper entitled "New Orleans business recovery in the aftermath of Hurricane Katrina" to study the decisions of shop owners to reopen business after Hurricane Katrina. The dataset contains 673 observations on 3 streets in New Orleans and can be used to estimate the spatial probit models and to replicate the findings in the paper.

Usage

```
data(Katrina)
```

Format

Katrina is a data frame with 673 observations on the following 15 variables:

`code` a numeric vector

`long` longitude coordinate of store

`lat` latitude coordinate of store

`street1` a numeric vector

`medinc` median income

`perinc` a numeric vector

`elevation` a numeric vector

`flood` flood depth (measured in feet)

`owntype` type of store ownership: "sole proprietorship" vs. "local chain" vs. "national chain" sesstatus socio-economic status of clientele (1-5): 1-2 = low #' status customers, 3 = middle, 4-5 = high status customers

sizeemp "small size" vs. "medium size" vs. "large size" firms

openstatus1 a numeric vector

openstatus2 a numeric vector

days days to reopen business

street 1=Magazine Street, 2=Carrollton Avenue, 3=St. Claude Avenue

Katrina is a data frame with 673 observations on the following 13 variables.

long longitude coordinate of store

lat latitude coordinate of store

flood_depth flood depth (measured in feet)

log_medinc log median income

small_size binary variable for "small size" firms

large_size binary variable for "large size" firms

low_status_customers binary variable for low socio-economic status of clientele

high_status_customers binary variable for high socio-economic status of clientele

owntype_sole_proprietor a binary variable indicating "sole proprietor" ownership type

owntype_national_chain a binary variable indicating "national_chain" ownership type

y1 reopening status in the very short period 0-3 months; 1=reopened, 0=not reopened

y2 reopening status in the period 0-6 months; 1=reopened, 0=not reopened

y3 reopening status in the period 0-12 months; 1=reopened, 0=not reopened

Details

The Katrina dataset contains the data found on the website before some of the variables are recoded. For example, the socio-economic status of clientele is coded as 1-5 in the raw data, but only 3 levels will be used in estimation: 1-2 = low status customers, 3 = middle, 4-5 = high status customers. Hence, with "middle" as the reference category, Katrina contains 2 dummy variables for low status customers and high status customers.

The dataset Katrina is the result of these recoding operations and can be directly used for model estimation.

Note

When definining the reopening status variables y1 (0-3 months), y2 (0-6 months), and y3 (0-12 months) from the days variable, the Matlab code ignores the seven cases where days=90. To be consistent with the number of cases in the paper, we define y1,y2,y3 in the same way: y1=sum(days < 90), y2=sum(days < 180 & days != 90), y3=sum(days < 365 & days != 90). So this is not a bug, its a feature.

Source

The raw data was obtained from the Royal Statistical Society dataset website and brought to RData format by Wilhelm and Godinho de Matos (2013).

References

LeSage et al. (2011) P. LeSage, R. K. Pace, N. Lam, R. Campanella and X. Liu. New Orleans business recovery in the aftermath of Hurricane Katrina. *Journal of the Royal Statistical Society A*, 174, 1007–1027, 2011.

Wilhelm and Godinho de Matos (2013) S. Wilhelm and M. Godinho de Matos. Estimating Spatial Probit Models in R. *The R Journal* 5, 130–143, 2013.

Examples

```
## Not run:
data(Katrina)
attach(Katrina)
table(y1) # 300 of the 673 firms reopened during 0-3 months horizon, p.1016
table(y2) # 425 of the 673 firms reopened during 0-6 months horizon, p.1016
table(y3) # 478 of the 673 firms reopened during 0-12 months horizon, p.1016
detach(Katrina)


# replicate LeSage et al. (2011), Table 3, p.1017
require(spdep)

# (a) 0-3 months time horizon
# LeSage et al. (2011) use k=11 nearest neighbors in this case
nb <- knn2nb(knearneigh(cbind(Katrina$lat, Katrina$long), k=11))
listw <- nb2listw(nb, style="W")
W1 <- as(as_dgRMatrix_listw(listw), "CsparseMatrix")

fit1_cond <- ProbitSpatialFit(y1 ~ flood_depth + log_medinc + small_size +
large_size +low_status_customers +  high_status_customers +
owntype_sole_proprietor + owntype_national_chain,
W=W1, data=Katrina, DGP='SAR', method="conditional", varcov="varcov")
summary(fit1_cond)

fit1_FL <- ProbitSpatialFit(y1 ~ flood_depth + log_medinc + small_size +
large_size +low_status_customers +  high_status_customers +
owntype_sole_proprietor + owntype_national_chain,
W=W1, data=Katrina, DGP='SAR', method="full-lik", varcov="varcov")
summary(fit1_FL)

fit1_cond_10nn <- ProbitSpatialFit(y1 ~ flood_depth+ log_medinc+ small_size+
large_size +low_status_customers +  high_status_customers +
owntype_sole_proprietor + owntype_national_chain,
W=W1, data=Katrina, DGP='SAR', method="conditional", varcov="varcov",
control=list(iW_CL=10))
summary(fit1_cond_10nn)

# (b) 0-6 months time horizon
# LeSage et al. (2011) use k=15 nearest neighbors
nb <- knn2nb(knearneigh(cbind(Katrina$lat, Katrina$long), k=15))
listw <- nb2listw(nb, style="W")
W2 <- as(as_dgRMatrix_listw(listw), "CsparseMatrix")
```

```
fit2_cond <- ProbitSpatialFit(y2 ~ flood_depth + log_medinc + small_size +
large_size + low_status_customers + high_status_customers +
owntype_sole_proprietor + owntype_national_chain,
W=W2, data=Katrina, DGP="SAR", method="full-lik", varcov="varcov")
summary(fit2_cond)

fit2_FL <- ProbitSpatialFit(y2 ~ flood_depth + log_medinc + small_size +
large_size + low_status_customers + high_status_customers +
owntype_sole_proprietor + owntype_national_chain,
W=W2, data=Katrina, DGP="SAR", method="full-lik", varcov="varcov")
summary(fit2_FL)

# (c) 0-12 months time horizon
# LeSage et al. (2011) use k=15 nearest neighbors as in 0-6 months
W3 <- W2
fit3_cond <- ProbitSpatialFit(y3 ~ flood_depth + log_medinc + small_size +
large_size + low_status_customers + high_status_customers +
owntype_sole_proprietor + owntype_national_chain,
W=W3, data=Katrina, DGP="SAR", method="conditional", varcov="varcov")
summary(fit3_cond)

fit3_FL <- ProbitSpatialFit(y3 ~ flood_depth + log_medinc + small_size +
large_size + low_status_customers + high_status_customers +
owntype_sole_proprietor + owntype_national_chain,
W=W3, data=Katrina, DGP="SAR", method="full-lik", varcov="varcov")
summary(fit3_FL)

# replicate LeSage et al. (2011), Table 4, p.1018
# SAR probit model effects estimates for the 0-3-month time horizon
effects(fit1_cond)

# replicate LeSage et al. (2011), Table 5, p.1019
# SAR probit model effects estimates for the 0-6-month time horizon
effects(fit2_cond)

# replicate LeSage et al. (2011), Table 6, p.1020
# SAR probit model effects estimates for the 0-12-month time horizon
effects(fit3_cond)

## End(Not run)
```

names.ProbitSpatial *Extract names of ProbitSpatial class.*

Description

Extract names of ProbitSpatial class.

Usage

```
## S3 method for class 'ProbitSpatial'
names(x, ...)
```

Arguments

`x` an object of class `ProbitSpatial`.

`...` ignored

Value

Returns the names of the `ProbitSpatial` object.

`predict.ProbitSpatial` *Spatial probit model predictions.*

Description

Predicts of a `ProbitSpatial` model on a set `X` of covariates. Works on both in-sample and out-of-sample using BLUP formula from Goulard et al. (2017)

Usage

```
## S3 method for class 'ProbitSpatial'
predict(
  object,
  X,
  type = c("link", "response", "binary"),
  cut = 0.5,
  oos = FALSE,
  WSO = NULL,
  ...
)
```

Arguments

`object` an object of class `ProbitSpatial`.

`X` a matrix of explanatory variables. If oos=TRUE, it may contain more observations than the dataset on which the model has been trained

`type` the type of output:

- `"link"` the value of the latent variable. Default
- `"response"` probability.
- `"binary"` binary 0/1 output.

`cut` the threshold probability for the `"binary"` type. Default is 0.5.

`oos` logical. If TRUE, out-of-sample predictions are returned.

`WSO` W matrix containing weights of in-sample and out-of-sample data. Observations must be ordered in such a way that the first elements belong to the in-sample data and the remaining ones to the out-of-sample data.

`...` ignored

Details

If `oos=FALSE`, the function computes the predicted values for #' the estimated model (same as `fitted`). Otherwise, it applies the BLUP #' formula of Goulard et al. (2017):

$$\hat{y} = (\hat{(y_S)}, \hat{(y_O)}),$$

where the sub-indexes S and O refer, respectively, to the in-sample and out-of-sample data. $\hat{y_S}$ corresponds to fitted values, while $\hat{y_O}$ is computed as follows:

$$\hat{y_O} = (I - \rho W)^{-1}(X\beta) - Q_{OO}^{-1} Q_{OS}(y_S - \hat{y_S}),$$

where Q is the precision matrix of $\Sigma = \sigma^2((I - \rho W)'(I - \rho W))^{-1}$. and the sub-indexes OO and OS refer to the corresponding block matrices.

Value

Returns a vector of predicted values for the set `X` of covariates if `oos=FALSE` or the best linear unbiased predictors of the #' set `XOS` if `oos=TRUE`.

References

Goulard et al. (2017) M. Goulard, T. Laurent and C. Thomas-Agnan. About predictions in spatial autoregressive models: optimal and almost optimal strategies. *Spatial Economic Analysis* 12, 304-325, 2017.

`ProbitSpatial-class` *Class of Spatial Probit Model.*

Description

Class of Spatial Probit Model.

Slots

`beta` numeric, the estimated parameters for the covariates.

`rho` numeric, the estimated spatial autocorrelation parameter.

`lambda` numeric, the estimated spatial error autocorrelation parameter.

`coeff` numeric, all estimated parameters.

`loglik` numeric, the likelihood associated to the estimated model.

`formula` `formula`.

`nobs` numeric, number of observations.

`nvar` numeric, number of covariates.

`y` numeric, vector of observed dependent variable.

`X` matrix, matrix of covariates.

`time` numeric, estimation time.

`DGP` character, DGP of the model (SAR, SEM or SARAR).

`method` character, estimation method ("`conditional`" or "`full-lik`").

`varcov` character, indicates the matrix used in the algorithm ("`varcov`" or "`precision`").

`W` SparseMatrix, the spatial weight matrix of y.

`M` SparseMatrix, the spatial weight matrix of the disturbances.

`iW_CL` numeric, the order of approximation used in the conditional method.

`iW_FL` numeric, the order of approximation used inside the likelihood function for the `full-lik` method.

`iW_FG` numeric, the order of approximation used inside the gradient functions for the `full-lik` method.

`reltol` numeric, the relative convergence tolerance.

`prune` numeric, the pruning for the gradient functions.

`env` an `environment` containing information for use in later function calls to save time.

`message` a integer giving any additional information or NULL.

`ProbitSpatialFit` *Fit a spatial probit model.*

Description

Approximate likelihood estimation of the probit model with spatial autoregressive (SAR), spatial error (SEM), spatial autoregressive with autoregressive disturbances (SARAR).

Usage

```
ProbitSpatialFit(formula,data,W,
         DGP='SAR',method="conditional",varcov="varcov",
         M=NULL,control=list())
```

Arguments

`formula` an object of class `formula`: a symbolic description of the model to be fitted.

`data` the data set containing the variables of the model.

`W` the spatial weight matrix of class "`dgCMatrix`".

`DGP` the data generating process of `data`: SAR, SEM, SARAR (Default is SAR).

`method` the optimisation method: `"conditional"` or `"full-lik"` (Defaul is `"conditional"`, see Details).

`varcov` the likelihood function is computed using the variance-covariance matrix (`"varcov"`) or the precision matrix (`"precision"`)? Default is `"varcov"`.

`M` the second spatial weight matrix for SARAR models. Same class as W.

`control` a list of control parameters. See Details.

Details

The estimation is based on the approximate value of the true likelihood of spatial probit models. The DGP of the spatial autoregressive model (SAR) model is the following

$$y = \rho W y + X\beta + \epsilon,$$

where the disturbances ϵ are iid standard normally distributed, W is a sparse spatial weight matrix and ρ is the spatial lag parameter. The variance of the error term is equal to $\Sigma = \sigma^2((I_n - \rho W)^{-1}((I_n - \rho W)^{-1})^t)$. The DGP of the spatial error model (SEM) is as follows

$$y = X\beta + u,$$

$$u = \rho W u + \epsilon,$$

where the disturbances ϵ are iid standard normally distributed, W is a sparse spatial weight matrix and ρ is the spatial error parameter. The variance of the error term is equal to $\Sigma = \sigma^2((I_n - \rho W)^{-1}((I_n - \rho W)^{-1})^t)$. The DGP of the spatial autoregressive model with autoregressive disturbances (SARAR) is as follows

$$y = \rho W y + X\beta + u,$$

$$u = \lambda M u + \epsilon,$$

where the disturbances ϵ are iid standard normally distributed, W and M are two sparse spatial weight matrix, while ρ and λ are the spatial lag and spatial error parameters, respectively. The variance of the error term is equal to $\Sigma = \sigma^2((I_n - \rho W)^{-1}(I_n - \lambda M)^{-1}((I_n - \lambda M)^{-1})^t((I_n - \rho W)^{-1})^t)$.

The approximation is inspired by the Mendell-Elston approximation of the multivariante normal probabilities (see References). It makes use of the Cholesky decomposition of the variance-covariance matrix Σ.

The `ProbitSpatialFit` command estimates the model by maximising the approximate log-likelihood. We propose two optimisation method:

`"conditional"`: it relies on a standard probit estimation which applies to the model estimated conditional on ρ.

`"full-lik"`: it minimises the full-log-likelihood using the analytical gradient functions (only available for SAR and SEM specification). The optimisation is performed by means of the `optim` function with `method = "BFGS"`.

In both cases a `"conditional"` estimation is performed. If `method="conditional"`, then `ProbitSpatialFit` returns the results of this first estimation. In case `method="full-lik"`, the function tries to improve the log-likelihood by means of a further exploration around the value of the parameters found by the conditional step. The conditional step is usually very accurate and particularly fast. The second step is more time consuming and does not always improve the results of the first step. We dissuade the user from using the full-likelihood method for sample sizes bigger than ten thousands, since the computation of the gradients is quite slow. Simulation studies reported in Martinetti and Geniaux (2017) prove that the conditional estimation is highly reliable, even if compared to the full-likelihood ones.

In order to reduce the computation time of the function `ProbitSpatialFit`, we propose a variant of the likelihood-function estimation that uses the inverse of the variance-covariance matrix (a.k.a. precision matrix). This variant applies to both the `"conditional"` and the `"full-lik"` methods and can be invoked by setting `varcov="precision"`. Simulation studies reported in Martinetti and Geniaux (2017) suggest that the accuracy of the results with the precision matrix are sometimes worst than the one with the true variance-covariance matrix, but the estimation time is considerably reduced.

The control argument is a list that can supply any of the following components:

`iW_CL` the order of approximation of $(I_n - \rho W)^{-1}$ used in the `"conditional"` method. Default is 6, while 0 means no approximation (it uses exact inversion of matrixes, not suitable for big sample sizes). See Martinetti and Geniaux (2017) for further references.

`iW_FL` the order of approximation of $(I_n - \rho W)^{-1}$ used in the computation of the likelihood function for the `"full-lik"` method. Default is 0, meaning no approximation.

`iW_FG` the order of approximation of $(I_n - \rho W)^{-1}$ used in the computation of the gradient functions for the `"full-lik"` method. Default is 0, meaning no approximation.

`reltol` relative convergence tolerance. It represents `tol` in `optimize` function for `method="conditional"` and `reltol` in `optim` function for `method="full-lik"`. Default is 1e-5.

`prune` the pruning value used in the gradients. Default is 0, meaning no pruning. Typacl values are around 1e-3 and 1e-6. They help reducing the estimation time of the gradient functions.

`silent` Default is TRUE.

Value

Return an object of class `ProbitSpatial`.

References

Mendell and Elston (1974) N. Mendell and R. Elston. Multifactorial qualitative traits: genetic analysis and prediction of recurrence risks. *Biometrics* 30, 41–57, 1974.

Martinetti and Geniaux (2017) D. Martinetti and G. Geniaux. Approximate likelihood estimation of spatial probit models. *Regional Science and Urban Economics* 64, 30-45, 2017.

Examples

```
n <- 1000
nneigh <- 3
rho <- 0.5
beta <- c(4,-2,1)
W <- generate_W(n,nneigh,seed=123)
X <- cbind(1,rnorm(n,2,2),rnorm(n,0,1))
colnames(X) <- c("intercept","X1","X2")
```

```
y <- sim_binomial_probit(W=W,X=X,beta=beta,rho=rho,model="SAR")
d <- as.data.frame(cbind(y,X))
mod <- ProbitSpatialFit(y~X1+X2,d,W,
       DGP='SAR',method="conditional",varcov="varcov")
```

residuals.ProbitSpatial

Extract spatial probit model residuals.

Description

Compute the residuals of an estimated `ProbitSpatial` model.

Usage

```
## S3 method for class 'ProbitSpatial'
residuals(object, ...)
```

Arguments

`object`	an object of class `ProbitSpatial`.
`...`	ignored

Value

Return a vector containing the generalised residuals of the `ProbitSpatial` model.

sim_binomial_probit

Simulate the dependent variable of a SAR/SEM/SARAR model.

Description

The function `sim_binomial_probit` is used to generate the dependent variable of a spatial binomial probit model, where all the data and parameters of the model can be modified by the user.

Usage

```
sim_binomial_probit(W,X,beta,rho,model="SAR",M=NULL,lambda=NULL,
sigma2=1,ord_iW=6,seed=123)
```

Arguments

W	the spatial weight matrix (works for "SAR" and "SEM" models).
X	the matrix of covariates.
beta	the value of the covariates parameters.
rho	the value of the spatial dependence parameter (works for "SAR" and "SEM" models).
model	the type of model, between "SAR", "SEM", "SARAR" (Default is "SAR").
M	the second spatial weight matrix (only if model is "SARAR").
lambda	the value of the spatial dependence parameter (only if model is "SARAR").
sigma2	the variance of the error term (Defaul is 1).
ord_iW	the order of approximation of the matrix $(I_n - \rho W)^{-1}$.
seed	to set the random generator seed of the error term.

Details

The `sim_binomial_probit` generates a vector of dependent variables for a spatial probit model. It allows to simulate the following DGPs (Data Generating Process): SAR

$$z = (I_n - \rho W)^{-1}(X\beta + \epsilon)$$

SEM

$$z = X\beta + (I_n - \rho W)^{-1}\epsilon$$

SARAR

$$z = (I_n - \rho W)^{-1}(X\beta + (I_n - \lambda M)^{-1}\epsilon)$$

where ϵ are independent and normally distributed with mean zero and variance `sigma2` (default is 1).

The matrix `X` of covariates, the corresponding parameters `beta`, the spatial weight matrix `W` and the corresponding spatial dependence parameter `rho` need to be passed by the user. Eventually, the same applies for `lambda` and `M` for the SARAR model.

The matrix $(I_n - \rho W)^{-1}$ is computed using the `ApproxiW` function, that can either invert $(I_n - \rho W)$ exactely, if `order_iW=0` (not suitable for `n` bigger than 1000), or using the Taylor approximation

$$(I_n - \rho W)^{-1} = I_n + \rho W + \rho^2 W^2 + \dots$$

of order `order_iW` (default is approximation of order 6).

Value

a vector of zeros and ones

See Also

`generate_W`, `ProbitSpatialFit`.

Examples

```
n <- 500
nneigh <- 3
rho <- 0.5
beta <- c(4,-2,1)
W <- generate_W(n,nneigh)
X <- cbind(1,rnorm(n,2,2),rnorm(n,0,1))
#SAR
y <- sim_binomial_probit(W,X,beta,rho,model="SAR") #SAR model
#SEM
y <- sim_binomial_probit(W,X,beta,rho,model="SEM") #SEM model
#SARAR
M <- generate_W(n,nneigh,seed=1)
lambda <- -0.5
y <- sim_binomial_probit(W,X,beta,rho,model="SARAR",M=M,lambda=lambda)
```

summary.ProbitSpatial *Spatial probit model summaries.*

Description

Print the results of a ProbitSpatial model.

Usage

```
## S3 method for class 'ProbitSpatial'
summary(object, covar = FALSE, ...)
```

Arguments

object	an object of class ProbitSpatial.
covar	should the statistics be computed with the matrix of variance of the parametes or not. Default is FALSE, hence Likelihood-ratio statistics are printed.
...	further arguments

Details

The summary function prints

Model Featurs on the model and dataset.

Time Estimation time.

Statistics Standard errors of the estimated parameters. If covar=TRUE, it uses the matrix of variance of the parameters, else the likelihood ratio test.

Accuracy Confusion Matrix and accuracy of the estimated model.

Value

This functions does not return any value.

`$,ProbitSpatial-method`

Extract from ProbitSpatial class.

Description

extract a slot from `ProbitSpatial` class object.

Usage

```
## S4 method for signature 'ProbitSpatial'
x$name
```

Arguments

`x`	an object of class `ProbitSpatial`.
`name`	of the slot.

Value

The content of the slot.

Index